Frontiers
in
Artificial
Intelligence
and
Applications

ADVANCES IN LOGIC, ARTIFICIAL INTELLIGENCE AND ROBOTICS

LAPTEC 2002

Edited by
Jair M. Abe
João I. da Silva Filho

IOS
Press
Ohmsha

VISIT…

LANZAROTE
Caliente.com

ADVANCES IN LOGIC, ARTIFICIAL INTELLIGENCE AND ROBOTICS

Frontiers in Artificial Intelligence and Applications

Series Editors: J. Breuker, R. López de Mántaras, M. Mohammadian, S. Ohsuga and W. Swartout

Volume 85

Recently published in this series:

Vol. 84, H. Fujita and P. Johannesson (Eds.), New Trends in Software Methodologies, Tools and Techniques
Vol. 83, V. Loia (Ed.), Soft Computing Agents
Vol. 82, E. Damiani et al. (Eds.), Knowledge-Based Intelligent Information Engineering Systems and Allied Technologies
Vol. 81, In production
Vol. 80, T. Welzer et al. (Eds.), Knowledge-based Software Engineering
Vol. 79, H. Motoda (Ed.), Active Mining
Vol. 78, T. Vidal and P. Liberatore (Eds.), STAIRS 2002
Vol. 77, F. van Harmelen (Ed.), ECAI 2002
Vol. 76, P. Sinčák et al. (Eds.), Intelligent Technologies - Theory and Applications
Vol. 75, I.F. Cruz et al. (Eds.), The Emerging Semantic Web
Vol. 74, M. Blay-Fornarino et al. (Eds.), Cooperative Systems Design
Vol. 73, H. Kangassalo et al. (Eds.), Information Modelling and Knowledge Bases XIII
Vol. 72, A. Namatame et al. (Eds.), Agent-Based Approaches in Economic and Social Complex Systems
Vol. 71, J.M. Abe and J.I. da Silva Filho (Eds.), Logic, Artificial Intelligence and Robotics
Vol. 70, B. Verheij et al. (Eds.), Legal Knowledge and Information Systems
Vol. 69, N. Baba et al. (Eds.), Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies
Vol. 68, J.D. Moore et al. (Eds.), Artificial Intelligence in Education
Vol. 67, H. Jaakkola et al. (Eds.), Information Modelling and Knowledge Bases XII
Vol. 66, H.H. Lund et al. (Eds.), Seventh Scandinavian Conference on Artificial Intelligence
Vol. 65, In production
Vol. 64, J. Breuker et al. (Eds.), Legal Knowledge and Information Systems
Vol. 63, I. Gent et al. (Eds.), SAT2000
Vol. 62, T. Hruška and M. Hashimoto (Eds.), Knowledge-Based Software Engineering
Vol. 61, E. Kawaguchi et al. (Eds.), Information Modelling and Knowledge Bases XI
Vol. 60, P. Hoffman and D. Lemke (Eds.), Teaching and Learning in a Network World
Vol. 59, M. Mohammadian (Ed.), Advances in Intelligent Systems: Theory and Applications
Vol. 58, R. Dieng et al. (Eds.), Designing Cooperative Systems
Vol. 57, M. Mohammadian (Ed.), New Frontiers in Computational Intelligence and its Applications
Vol. 56, M.I. Torres and A. Sanfeliu (Eds.), Pattern Recognition and Applications
Vol. 55, G. Cumming et al. (Eds.), Advanced Research in Computers and Communications in Education
Vol. 54, W. Horn (Ed.), ECAI 2000
Vol. 53, E. Motta, Reusable Components for Knowledge Modelling
Vol. 52, In production
Vol. 51, H. Jaakkola et al. (Eds.), Information Modelling and Knowledge Bases X
Vol. 50, S.P. Lajoie and M. Vivet (Eds.), Artificial Intelligence in Education

ISSN: 0922-6389

Advances in Logic, Artificial Intelligence and Robotics

LAPTEC 2002

Edited by

Jair Minoro Abe

SENAC – College of Computer Science and Technology, São Paulo, Brazil

and

João Inácio da Silva Filho

SENAC – College of Computer Science and Technology, São Paulo, Brazil

IOS
Press
Ohmsha

Amsterdam • Berlin • Oxford • Tokyo • Washington, DC

© 2002, The authors mentioned in the Table of Contents

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, without prior written permission from the publisher.

ISBN 1 58603 292 5 (IOS Press)
ISBN 4 274 90545 4 C3055 (Ohmsha)
Library of Congress Control Number: 2002112411

Publisher
IOS Press
Nieuwe Hemweg 6B
1013 BG Amsterdam
The Netherlands
fax: +31 20 620 3419
e-mail: order@iospress.nl

Distributor in the UK and Ireland
IOS Press/Lavis Marketing
73 Lime Walk
Headington
Oxford OX3 7AD
England
fax: +44 1865 75 0079

Distributor in the USA and Canada
IOS Press, Inc.
5795-G Burke Centre Parkway
Burke, VA 22015
USA
fax: +1 703 323 3668
e-mail: iosbooks@iospress.com

Distributor in Germany, Austria and Switzerland
IOS Press/LSL.de
Gerichtsweg 28
D-04103 Leipzig
Germany
fax: +49 341 995 4255

Distributor in Japan
Ohmsha, Ltd.
3-1 Kanda Nishiki-cho
Chiyoda-ku, Tokyo 101-8460
Japan
fax: +81 3 3233 2426

LEGAL NOTICE
The publisher is not responsible for the use which might be made of the following information.

PRINTED IN THE NETHERLANDS

Preface

It constitutes a great honor for us to present the approved and invited papers of the 3rd Congress of Logic Applied to Technology - LAPTEC 2002, held in São Paulo, Brazil, from November 11 to 13, 2002. Logic (Classical and Non-Classical) is being increasingly connected with almost every scientific discipline and human activity. In this volume we have emphasized its role in Artificial Intelligence, Robotics, Informatics in general, Technology, and correlated themes.

LAPTEC 2002 had as Chairs:

General Chairs:

Jair Minoro Abe (Brazil) (Chair)
João Inácio da Silva Filho (Brazil) (Vice-Chair)

Organizing Committee:

Alexandre Scalzitti (Germany), Cláudio Rodrigo Torres (Brazil - Chair), Flávio Shigeo Yamamoto (Brazil), Marcos Roberto Bombacini (Brazil - Vice-Chair), and Neli Regina S. Ortega (Brazil).

On behalf of the Organizing Committee, we would like to express our gratitude to the members of the following committees:

Honorary Chairs:

Edgar G. Lopez-Escobar (U.S.A), Eduardo Massad (Brazil), Germano Lambert Torres (Brazil), Heinz-Dieter Ebbinghaus (Germany), Kiyoshi Iséki (Japan), Lotfi A. Zadeh (U.S.A.), Newton C.A. da Costa (Brazil), and Patrick Suppes (U.S.A.).

Scientific Committee:

Atsuyuki Suzuki (Japan), Bráulio Coelho Ávila (Brazil), Daniel Dubois (Belgium), Dietrich Kuske (Germany), Edgar G. Lopez-Escobar (U.S.A), Eduardo Massad (Brazil), Germano Lambert Torres (Brazil), Germano Resconi (Italy), Guo-Qiang Zhang (U.S.A.), Heinz-Dieter Ebbinghaus (Germany), Helmut Thiele (Germany), Hiroakira Ono (Japan), João Inácio da Silva Filho (Brazil), John Meech (Canada), Kazumi Nakamatsu (Japan), Kiyoshi Iséki (Japan), Lotfi A. Zadeh (U.S.A.), Manfred Droste (Germany), Marcel Guillaume (France), Maria Carolina Monard (Brazil), Mineichi Kudo (Japan), Miyagi Kiyohiro (Japan), Nelson Fávilla Ebecken (Brazil), Newton C.A. da Costa (Brazil), Paulo Veloso (Brazil), Patrick Suppes (U.S.A.), Seiki Akama (Japan), Setsuo Arikawa (Japan), Sheila Veloso (Brazil), Ricardo Bianconi (Brazil), Tadashi Shibata (Japan), Tetsuya Murai (Japan), and Tsutomu Date (Japan).

Also our special gratitude to the following additional scholars who helped us in refereeing papers: João Pereira da Silva – DCC UFRJ (Brazil), Rosa Esteves – UERJ (Brazil), Maria das Graças Volpe Nunes – USP (Brazil), Alneu de Andrade Lopes – USP (Brazil), Gustavo Batista – USP (Brazil), Claudia Martins – USP (Brazil), José Augusto Baranauskas – USP (Brazil), Claudia Milaré – USP (Brazil), Roseli Francelin – USP (Brazil), João Batista Neto – USP (Brazil), Adriano Donizete Pila – USP (Brazil), Alexandre Rasi Aoki – UNIFEI (Brazil), Cláudio Inácio de Almeida Costa – UNIFEI (Brazil), and Luiz Eduardo Borges da Silva – UNIFEI (Brazil).

We would like to thank the numerous sponsors, particularly the SENAC – College of Computer Science and Technology, São Paulo, Brazil. We would also like to acknowledge the following entities: FAPESP, IEEE, CNPq, Institute For Advanced Studies – University of São Paulo, Universtity of São Paulo – Campus São Carlos, Himeji Institute of Technology – Japan, Shizuoka University – Japan, Teikyo Heisei University – Japan, UNIFEI – Universidade Federal de Itajubá – Brazil, Federal University of Rio de Janeiro – Brazil, Sociedade Brasileira de Computação, Sociedade Brasileira para o Progresso da Ciência, ABJICA – Brazil, Hokkaido University – Japan, The University of Britsh Columbia – Canada, Universität Dortmund – Germany, University of Liége – Belgium, Stanford University – U.S.A., University of the Ryukyus – Japan, Centrais Elétricas do Norte do Brasil S.A– Eletronorte, FURNAS Centrais Elétricas S.A, and IOS Press, the publisher of this Proceedings.

Our undying gratitude again for all these gifted people, responsible for the success of LAPTEC 2002.

Jair Minoro Abe
João Inácio da Silva Filho
Chair and Vice-Chair – LAPTEC 2002

Contents

Preface v

Retriever Prototype of a Case Based Reasoning: A Study Case, *H.G. Martins and G. Lambert-Torres* 1
Dynamic Compaction Process of Metal Powder Media within Dies, *K. Miyagi, Y. Sano, T. Sueyoshi and Z. Nakao* 9
Automated Theorem Proving for Many-sorted Free Description Theory Based on Logic Translation, *K. Nakamatsu and A. Suzuki* 17
Annotated Logic and Negation as Failure, *K. Nakamatsu and A. Suzuki* 29
Multi-agent System for Distribution System Operation, *A.R. Aoki, A.A.A. Esmin and G. Lambert-Torres* 38
ArTbitrariness: Putting Computer Creativity to Work in Aesthetic Domains, *A. Moroni, J. Manzolli and F.J. Von Zuben* 46
An Overview of Fuzzy Numbers and Fuzzy Arithmetic, *F. Gomide* 55
The Brain and Arithmetic Calculation, *F.T. Rocha and A.F. Rocha* 62
Evolving Arithmetical Knowledge in a Distributed Intelligent Processing System, *A.F. Rocha and E. Massad* 68
Meme–Gene Coevolution and Cognitive Mathematics, *E. Massad and A.F. Rocha* 75
Neuronal Plasticity: How Memes Control Genes, *A. Pereira Jr.* 82
The Influence of Heterogeneity in the Control of Diseases, *L.C. de Barros, R.C. Bassanezi, R.Z.G. de Oliveira* 88
Paraconsistent Logics viewed as a Foundation of Data Warehouses, *S. Akama and J.M. Abe* 96
Visualization of Class Structures using Piecewise Linear Classifiers, *H. Tenmoto, Y. Mori and M. Kudo* 104
Design of Tree Classifiers using Interactive Data Exploration, *Y. Mori and M. Kudo* 112
Clustering Based on Gap and Structure, *M. Kudo* 120
Tables in Relational Databases from a Point of View of Possible-Worlds-Restriction, *T. Murai, M. Nakata and Y. Sato* 126
On Some Different Interpretations of the Generalized Modus Ponens using Type-2 Fuzzy Sets, *H. Thiele* 134
Paraconsistent Knowledge for Misspelling Noise Reduction in Documents, *E.L. dos Santos, F.M. Hasegawa, B.C. Ávila and C.A.A. Kaestner* 144
Automata with Concurrency Relations — A Survey, *M. Droste and D. Kuske* 152
Learning with Skewed Class Distributions, *M.C. Monard and G.E.A.P.A. Batista* 173
An Enlargement of Theorems for Sentential Calculus, *S. Tanaka* 181
A Real-time Specification Language, *F.N. do Amaral, E.H. Haeusler and M. Endler* 194
Defuzzification in Medical Diagonis, *J.C.R. Pereira, P.A. Tonelli, L.C. de Barros and N.R.S. Ortega* 202
Fuzzy Rules in Asymptomatic HIV Virus Infected Individuals Model, *R.S. da Motta Jafelice, L.C. de Barros, R.C. Bassanezi and F. Gomide* 208
Categorical Limits and Reuse of Algebraic Specifications, *I. Cafezeiro and E.H. Haeusler* 216
Constructive Program Synthesis using Intuitionist Logic and Natural Deduction, *G.M.H. Silva, E.H. Haeusler and P.A.S. Veloso* 224

An Agent-oriented Inference Engine Applied for Supervisory Control of Automated Manufacturing Systems, *J.M. Simão and P.C. Stadzisz* 234
LTLAS: a Language Based on Temporal Logic for Agents Systems Specification, *N.F. Mendoza and F.F. Ramos Corchado* 242
Fuzzy Identification of a pH Neutralization Process, *R.A. Jerônimo, I.A. Souza and E.P. Maldonado* 250
A Fuzzy Reed–Frost Model for Epidemic Spreading, *T.E.P. Sacchetta, N.R.S. Ortega, R.X. Menezes and E. Massad* 258

Invited Talks – Abstracts

Data Mining in Large Base using Rough Set Techniques, *G. Lambert-Torres* 267
An Automaton Model for Concurrent Processes, *M. Droste* 268
It is a Fundamental Limitation to Base Probability Theory on Bivalent Logic, *L.A. Zadeh* 269
Polysynthetic Class Theory for the 21st Century, *E.G.K. Lopez-Escobar* 272
Development of a Fuzzy Genetic System for Data Classification, *N. Ebecken* 273
Why the Problem P = NP is so Difficult, *N.C.A. da Costa* 274
The Importance of Evaluating Learning Algorithms during the Data Mining Process, *M.C. Monard and G.E.A.P.A. Batista* 275

Author Index 277

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Retriever Prototype of a Case Based Reasoning: A Study Case

Helga Gonzaga Martins **Germano Lambert- Torres**
Federal University at Itajubá
Av. BPS 1303 – Itajubá – 37500-000 – MG – Brazil – germano@iee.efei.br

Abstract – This paper presents the results from application back up of diagnostic decision supported by a Case Based Reasoning (CBR), with the presentation of a retriever prototype used for cases recovering from a specialist domain. Its aim is to recover, from a memory of cases, the most adequate case for a new situation and to suggest the solution for the new case.

1 - Introduction

The paradigm Case Based Reasoning (CBR) presume the existence of a memory where the already solved cases are stored; uses these cases, through recovering, for helping in the resolution or interpretation of new problems; and promotes the learning, allowing that new cases (newly solved or newly spell-out) be added to the memory [1].

One CBR uses previous cases as far for evaluation, justification or interpretation of proposed solutions (interpretative CBR), as for proposing solutions for new problems (problem solving CBR) [2].

Case based reasoning have been opening new fields on computer back up regarding decision problems of a bad structure.

This technique have been used as decision back up in many knowledge domains such as: planning, projects, diagnostics, with a better performance than other reasoning systems; as example we mention PROTOS and CASEY as referred in [1], AIRQUAP in [3], a water sewage treatment system developed by [4] and e direct current motor repair system named TRAAC [5].

The CBR system purpose is to recover from its memory the most similar case regarding to the new, suggest the solution or one adaptation of this as a solution for the new situation.

The usefulness of the old cases is made by the similarity access of a new case with the old one.

The central methodology of the retriever prototype is the similarity determination of a new case with all the previous cases. The similarities are set up through combination functions (matching) and through the characteristics of the new case with all the previous ones. In the next sections we will show how to handle with similarities and the matching function, so as the architecture of the retriever prototype.

2 – Similarities

A case may be considered as a scheme comprising one set of pairs of attribute values, that is to say, *descriptions* [1]. For example, in one decision scenery for credit evaluation, one loans manager access several pairs of attribute values, namely, the "candidate type" has a "medium" value. He set up a similarity combination of the new case scheme with a previous case scheme. This combination is made in two steps:

1) To find the *similarity* of new case scheme with a previous case scheme along the descriptions;
2) To find the *global similarity* through combination functions [5].

The similarity between the new case and a previous case along the descriptions have been find using a *domain knowledge* made up of rules of heuristic combinations and specific domain [6]. As an example, a combination used for the determination of a description "oven ember color" with one *orange* value is more similar to the description of *red* value.

The global similarity of a new case with previous cases is found through *Combination Functions* adding the similarities along the descriptions, the used function in this work is the *Modified Cosine Function,* shown as follows.

3 – Function Matching Modified Cosine

According [5] the function modified cosine is represented as follows:

$$\text{Gcas}=\frac{\sum_{i=1}^{m}\omega_i^n\omega_i^{p_k}\left(1-\left(\frac{(x_i-y_i)}{R_i}\right)\right)}{\sqrt{\sum_{i=1}^{m}\left(\omega_i^n\right)^2\sum_{i=1}^{m}\left(\omega_i^{p_k}\right)^2}}$$

for: i = 1,...m (descriptions) and
for: k = 1,...r (previous cases)
Com:

$1-\left(\frac{(x_i-y_i)}{R_i}\right)$ ≡ denotes the similarity in the i- th description of the new case and a previous case;

ω_i^n ≡ weight of i- th description on weight vector from new case;

$\omega_i^{p}{}_k$ ≡ weight of i- th description on weight vector from previous case.

The combination modified cosine determine the global similarity, named "*Matching degree*", **Gcas**, between two cases by comparison of terms frequency, that is to say, the description weight from the new case and the terms weight from previous case.

The function measures the cosine of the angle between the vectors weight of the new and previous cases, which cosine is weighed by the similarity degree along the space m-dimensions of descriptions.

The denominator terms of equation above normalize the vectors weight by the determinations of its Euclidian lengths.

The similarity function is based in the pertinence (weight) of description values for the diagnostic. The similarity between the value of the present description from the new case and the value of same present description in the previous case of memory is taken as being the difference between the unit and a rate between the weights that each one of these values have for the diagnostic of the case in memory, with the extension value of the description scale.

The importance determination of one description is set up in one scale as per Figure 1:

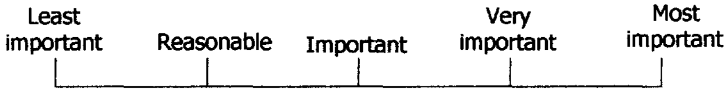


Figure 1 –Determination of Description Importance.

For example, presuming that the description of one specific system be the *temperature* and the determination of its importance pass through one scale which extension is defined as follow in Figure 2:

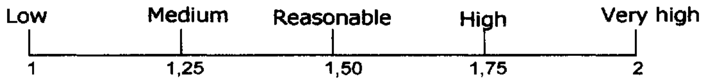


Figure 2 –Temperature Quality Values.

The similarity along the description may be computed, for example, for one value of *Very High Temperature* (2) in combination with a value of *Medium Temperature* (1,25) like:

$$\text{Similarity along description} = 1 - (2{,}0 - 1{,}25) / 2{,}0$$
$$\text{Similarity} = 0{,}625$$

4 -Retrieve Process

The recovery process in Case Based Reasoning – CBR, comprises experience of past solutions stored into a memory known as *cases*. This technique aims to recover the most useful previous cases toward the solution of the new decision take problem and to ignore the irrelevant previous cases.

The cases recovery works in the following way, as set up in Figure 3: based upon the *description* of the new decision take problem (new case) the basic case is searched by the previous cases starting from a decision back up. The *search* is made based upon similarities [7]. The previous cases get through the combination function (matching degree) and are ordered in decreasing way regarding the matching degree. The combination function determines the similarity degree of the useful potential from previous cases with one new case.

The retriever prototype requires the whole evidence body supplied by memory, that is to say, it requires that the entry case matching degree be computed against all the memory cases.

In Figure 4 we see the Retriever Prototype architecture. For each memory case a matching function is defined between the new case and the same. This function computes the *Belief Function* in favor of the diagnostic of this case for the case of entry.

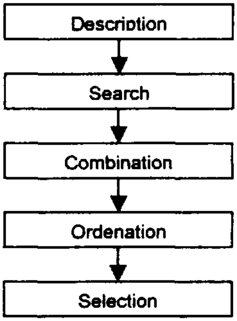

Description
Search
Combination
Ordenation
Selection

Figure 3 – Recover Components of one CBR

The previous cases determined by search may be combined by **Gcas** and arranged in global similarity decreasing order.

In the diagnostic domain, it is usual that the cases in which the same diagnostic happens by symptoms groups or different characteristics.

"The most adequate diagnostic" supposed for the entry situation, is the one that presents the major evidence to its favor, in other words, the one that has a bigger Belief Degree, computed by the Matching Degree, **Gcas**.

"The most adequate case" suggested is the one among all the cases from the class of selected diagnostic, that have the bigger Matching degree **Gcas**. The cases that belong to that diagnostic class may be of help for the new problem solution.

The advantage in recovering the case is that it is possible to find in it information that were useful for the solution of previous problems which may help for solving the new case.

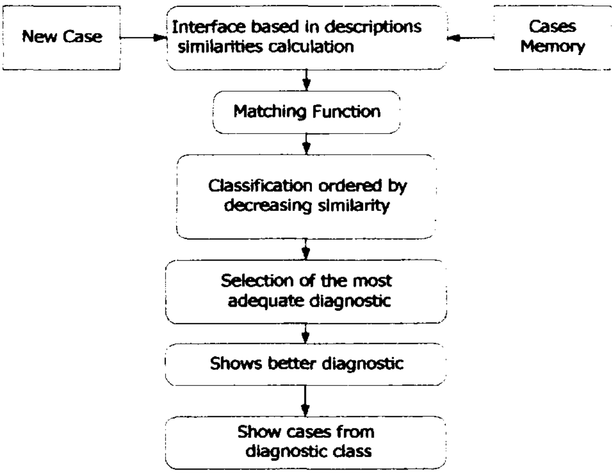

New Case
Interface based in descriptions similarities calculation
Cases Memory
Matching Function
Classification ordered by decreasing similarity
Selection of the most adequate diagnostic
Shows better diagnostic
Show cases from diagnostic class

Figure 4 –Retriever Prototype Architecture.

5 – Reasoning Based in Cases for Diagnostics

Our CBR is used for diagnostic determination for the operation of a cement kiln [8].

The kiln operator is aim at production of *High Quality* cement, which is achieved basically by the cement conditions.

5.1 – Decision Domain

The operator target is the value control of two kiln parameters, named *Rotation Velocity (RV)* and *Temperature (T)*. These parameters values are set up by the operator considering four other parameters that are: *Granulation (G)*, *Viscosity (V)*, *Color (C)*, and *pH Level*. A decision table where G, V, C and pH are "conditioning attributes" and RV and T are "decision attributes" may describe these actions. The condition attribute values when combined, correspond to cement specific quality produced in the kiln, and for each one of these attributes adequate actions are expected to provide high quality. All attributes and its values are listed according Table 1.

Table 1 – Specification of Descriptions and its Extensions.

Attributes	Descriptions	Extension of Importance Scale
Condition	a – Granular	0 - 3
Condition	b – Viscosity	0 - 3
Condition	c – Color	0 - 3
Condition	d - pH Level	0 - 3
Decision	e - Rotational Velocity	0 - 3
Decision	f – Temperature	0 - 3

In Table 2 are shown the possible diagnostics and its identification according to the descriptions *rotational velocity* and *temperature.*

Table 2 - Diagnostics and its extensions

Diagnostic	e	f
D1	1	3
D2	0	3
D3	1	2
D4	1	1

5.2 –Knowledge Base

The knowledge base for CBR evaluation comprises 13 cases and is related to descriptions and with their diagnostics, as shown in Table 3:

Table 3 – Knowledge Base

Cases	a	b	c	d	Diagnostic
CASE 01	2	1	1	1	D1
CASE 02	2	1	1	0	D1
CASE 03	2	2	1	1	D1
CASE 04	1	1	1	0	D2
CASE 05	1	1	1	1	D2
CASE 06	2	1	1	2	D3
CASE 07	2	2	1	2	D3
CASE 08	3	2	1	2	D3
CASE 09	3	2	2	2	D4
CASE 10	3	3	2	2	D4
CASE 11	3	3	2	1	D4
CASE 12	3	2	2	1	D4
CASE 13	3	0	2	1	D4

5.3 – Program Results for Retriever Prototype

The Retriever Prototype was tested for several New Cases, the results of the first New Case- [2 2 2 1] will be presented, that is to say, being the more simple Knowledge Base. For the New Case - [2 1 0 2], the used Knowledge Base already has a bigger Previous Cases memory, once its knowledge is more comprising.

5.3 1 –New Case - [2 2 2 1]

CASES RECOVERING ANALYSYS

NEW CASE: [2 2 2 1]
Case found in Knowledge Base: 0
No Case Found

Calculation Process Begining

ORDER	Gcas	CASE	DIAGNOSTIC
1	0.9063	3	D1
2	0.8498	12	D4
3	0.8205	7	D3
4	0.8070	9	D4
5	0.8037	1	D1
6	0.7549	2	D1
7	0.7518	11	D4
8	0.7252	10	D4
9	0.7016	6	D3
10	0.6973	8	D3
11	0.6934	5	D2
12	0.6671	13	D4
13	0.6405	4	D2
14	0.6138	14	D4

New Knowledge Base:
Columns 1 through 14

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14
2	2	2	1	1	2	2	3	3	3	3	3	3	3
1	1	2	1	1	1	2	2	2	3	3	2	0	1
1	1	1	1	1	1	1	1	2	2	2	2	2	1
1	0	1	0	1	2	2	2	2	2	1	1	1	1

Column 15

C15
2
2
2
1

DIAGNOSTICS BELIEF DEGREE:
Cr(D1)= 8.216099e-001
Cr(D2)= 6.669439e-001
Cr(D3)= 7.398203e-001
Cr(D4)= 7.358005e-001

MORE ADEQUATE DIAGNOSTIC FOR NEW CASE:
Bigger Belief Degree: 8.216099e-001

Bigger Belief Degree Diagnostic: D 1
Cases from More Adequate Diagnostic: C1, C2, C3

PREVIOUS CASE MORE ADEQUATE TO THE NEW CASE:
Bigger Matching Degree: 9.062933e-001
Case with bigger Matching Degree: C3

5.3 2 -New Case - [2 1 0 2]

CASE RECOVERY ANALYSYS
NEW CASE:[2 1 0 2]
Case found in Knowledge Base: 0
No Case Found

Calculation Process Begining

ORDER	Gcas	CASE	DIAGNOSTIC
1	0.9487	6	D3
2	0.8629	7	D3
3	0.7979	1	D1
4	0.7333	8	D3
5	0.7276	18	D3
6	0.7027	3	D1
7	0.6804	2	D1
8	0.6789	9	D4
9	0.6163	15	D1
10	0.6111	5	D2
11	0.6094	14	D4
12	0.5883	10	D4
13	0.5451	19	D2
14	0.5238	12	D4
15	0.4751	13	D4
16	0.4491	4	D2
17	0.4402	11	D4
18	0.2733	16	D4
19	0.1784	17	D4

New Knowledge Base:

Columns 1 through 14

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14
2	2	2	1	1	2	2	3	3	3	3	3	3	3
1	1	2	1	1	1	2	2	2	3	3	2	0	1
1	1	1	1	1	1	1	1	2	2	2	2	2	1
1	0	1	0	1	2	2	2	2	2	1	1	1	1

Columns 15 through 20

C15	C16	C17	C18	C19	C20
2	1	1	2	1	2
2	3	3	3	1	1
2	3	3	0	2	0
1	1	0	2	3	2

DIAGNOSTICS BELIEF DEGREE:

Cr(D1)= 6.993502e-001
Cr(D2)= 5.350826e-001

```
Cr(D3)= 8.181133e-001
Cr(D4)= 4.709396e-001

MORE ADEQUATE DIAGNOSTIC FOR NEW CASE:

Bigger Belief Degree: 8.181133e-001
Diagnostic of Bigger Belief Degree: D 3
Cases of more adequate Diagnostic: C6, C7, C8, C18

PREVIOUS CASE MORE ADEQUATE FOR NEW CASE:

Bigger Matching Degree: 9.486833e-001
Case with bigger Matching Degree: C6
```

6 – Summary and future works

The basic idea in this work it is the knowledge retrieve process, in such a way that this knowledge be stored so as to allow us to simulate future actions in the diagnostics determination.

We have presented architecture for a Retriever Prototype of cases from one CBR.

As an additional and immediate work we pretend to involve special proprieties into Belief Function that makes the selection of the more adequate diagnostic for an entry situation and, consequently, the recovery of the more adequate case and of all cases belonging to this diagnostic class, which also may be of help for the new problem solution. These special proprieties are related to the Two Values-LPA2v Annotated Paraconsistent Logic, for the purpose of uniting two major areas: Case Based Reasoning and Paraconsistent Logic, according to [9] and [10].

Refferences

[1] Kolodner, Janet L., *"Case- Based Reasoning"*, San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1993.

[2] Kolodner, Janet L. and Leake, D. B.; *"A Tutorial introduction to case- based reasoning"*, in Case Based Reasoning: Experiences, Lessons, and Future Directions, D. B. Leake, Ed. Menlo Park, CA: AAAI Press, 1996.

[3] Lekkas, G. P., Avouris, N. M. & Viras, L. G., *"Case- Based Reasoning im Environmental Monitoring Applications"*, Applied Artificial Intelligence, 8: 359-376, 1994.

[4] Krovvidy, S. & Wee, W. G., *"Wastewater Treatment Systems from Case- Base Reasoning"*, Machine Learning, vol.10: 341-363, 1993.

[5] Gupta, K. M. and Montazemi, A. R., *"Empirical Evaluation of Retrieval in Case- Based Reasoning Systems Using Modified Cosine Matching Function"*, IEEE Transactions on Systems, man and cybernetics - Part A: Systems and Humans, vol. 27, nº 5, september, 1997.

[6] Porter, B. W., Bariess, R. & Holte R. C., *"Concept learning in weak theory domains"*, Artificial Intelligence, vol. 45, nº 1-2,pp. 229-264, Sept. 1990.

[7] Bariess, R. and King, J. A., *"Similarity assessment in case- based reasoning"*, im Proc. DARPA Workshop Case- Based Reasoning, San Mateo, CA: Morgan Kaufmann, pp. 67- 71, 1989.

[8] Sandness, G.D., *"A Parallel Learning System"*, CS890 Class Paper, Carnegie-Mellon University, pp.1-12, 1986.

[9] Gonzaga, Helga, Costa, C. I. A., Lambert- Torres, Germano, *"Generalization of Fuzzy and Classic Logic in NPL2v"*, Advances in Systems Science: Measurement, Circuits and Control - Electrical and Computer Engineering Series, Ed. N. E. Mastorakis and L. A. Pecorelli- Peres,WSES, pp. 79-83 , 2001.

[10] Lambert-Torres, G., Costa, C. I. A., Gonzaga, Helga, *"Decision- Making System based on Fuzzy and Paraconsistent Logics"*, Logic, Artificial Intelligence and Robotics - Frontiers in Artificial Intelligence and Applications, LAPTEC 2001, Ed. Jair M. Abe and João I. da Silva Filho – IOS Press, pp. 135-146 , 2001

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Dynamic Compaction Processes of Metal Powder Media Within Dies

By
K.Miyagi[†], Y.Sano[††], T.Sueyoshi and Z.Nakao[†††]
[†] ASME & JSME Member, Okinawa, Japan, [††] Kobe University Mercantile Marine, Kobe, Japan
[†††] Faculty of Engineering, University of the Ryukyus, Okinawa, Japan

Abstract. Dynamic compaction of a copper powder medium was studied through both theoretical analysis and experiments. By comparison of the theoretical and experimental data obtained, it was hoped to provide further useful experimental information.

Introduction

In this study a porous metal powder medium in a die is assumed to be a uniform continuum consisting of solid powder particles and air [5]. The pressure-density curve of the powder is known to be convex and a shock wave propagates in the porous metal powder medium when the powder is subjected to compaction [1]. In general, the shock wave propagating in a powder medium under jump conditions is treated with the use of the equations for the constitution of the powder, in addition to the Rankine-Hugoniot relation, also known as the conservation laws of momentum and mass [2].

In the study the static pressure density relation is used as the constitutive equation for the metal powder medium within the die, and elastic recovery is neglected in order to simplify the treatment of the reflection wave. For the experiment, a new measuring method was developed, and the dynamic behavior of the medium under compaction was examined by continuum [2,4]. The powder particles were assumed to move only in the direction of compaction. The medium in the die, which is finitely long and effectively infinitely long, was compacted [2,3,4] by the direct impact of the punch, and theoretical and experimental studies were made with regard to the data obtained.

In the experiment the copper powder medium in the die was marked on one side with fine lines of aluminum powder. The change of the interval of the lines was filmed with a high-speed camera, and the behavior of the powder and propagation of the shock wave in the powder medium were observed. It was found that in an effectively infinitely long powder medium, the powder particles show the same behavior as the theory predicts, and the observed densities increase as the die wall friction increases with time. It was also confirmed that in a finitely long powder medium, the powder particles act as predicted, and the theoretical results agree fairly well with the experimental results, and the compact has an approximately uniform density, which is affected somewhat by the die wall friction [6].

Basic Equation

Effectively Infinitely Long Metal Powder Medium in the Die

Let the x-axis be along the direction of the powder in the die and let it satisfy $0 < x < \infty$; in Fig.1 a small element along the x-axis between the points I and II of distance dx is taken and let A be the inner cross-sectional area of the die. We will consider that at $x = 0$ the punch with mass M impacts the powder surface with initial velocity v_0. Thus, when the mass impacts the surface of the powder medium, $x = 0$ and the velocity is v_0. We assume that the shock wave reaches the point I at some time t, and then it reaches the point II in time dt, and that the medium pressure, density and particle velocity are denoted, respectively, by p, ρ and v behind the shock wave front and by $\bar{p}$, $\bar{\rho}$ and $\bar{v}$ in front of it, and also that the velocity of the shock wave is given by c. We further assume that at time $t + dt$, points I and II move, respectively, to points I' and II', and that the wave front reaches the points II'. Then $\mathrm{I\,II}' = c\,dt$, $\mathrm{I\,I}' = v\,dt$, $\mathrm{II\,II}' = v\,dt$ and thus the infinitesimal element I II moves to $\mathrm{I}'\ \mathrm{II}'$. The state variables of the elements are transformed from $\bar{p}$, $\bar{\rho}$ and $\bar{v}$ to p, ρ and v. Consequently, from the law of conservation of mass,

$$\bar{\rho}(c-\bar{v}) = \rho(c-v) \tag{1}$$

From the laws of conservation of momentum and impulse, we have

$$p-\bar{p} = \bar{\rho}(c-\bar{v})(v-\bar{v}) \tag{2}$$

Before the impaction, the pressure and particle velocity of this powder satisfies $\bar{p} = \bar{v} = 0$. By letting $\bar{\rho} = \rho_0$, in equations (1) and (2), we get

$$\rho_0 c = \rho(c - v) \tag{3}$$

$$p = \rho_0 c v \tag{4}$$

Although the powder pressure gradually decreases with time behind the wave front because the punch velocity decreases due to powder resistance, the density does not change during unloading if the elastic recovery is ignored. Therefore, the density at each point in the area remains constant, which is determined by the pressure at the point immediately after the passing of the shock wave front. Thus, the element between the shock wave front and the rigid body (i.e. the punch) moves at the same velocity as the body. In figure 1, from the law of conservation of momentum of the medium to the left hand side of point I and the punch, we obtain

$$\begin{aligned} -A\bar{p}\,dt &= (M + m + dm)(v + dv) - (M + m)v - dm\,\bar{v} \\ &= (M + m)dv + dm(v - \bar{v}) \end{aligned} \tag{5}$$

where A is the cross-sectional area as before, m is powder mass between the shock wave front and the punch, and dm is the increment of m during dt given by $dm = \rho A(c - v)dt$.
Thus, equation (5) becomes

$$(M + m)\frac{dv}{dt} + \rho A(c - v)(v - \bar{v}) + A\bar{p} = 0 \tag{5'}$$

Further, by equations (1), (2) and (3), we obtain

$$(M + m)\frac{dv}{dt} + Ap = 0 \tag{6}$$

Since $\bar{p} = \bar{v} = 0$, from equation (5), we get $d\{(M + m)v\} = 0$, and hence,

$$(M + m)v = \text{Constant} \tag{7}$$

Since $m = 0$, $v = v_0$ immediately after the impaction, we have

$$v = \frac{Mv_0}{M + m} \tag{8}$$

During the passing of the wave front, the medium mass at x before deformation satisfies

$$m = \bar{\rho} A x \tag{9}$$

Consequently, the particle velocity at the moment v is given by

$$v = \frac{Mv_0}{M + \rho_0 A x} \tag{10}$$

We will use a static pressure-density relation. Although there are many equations proposed for the relation, the equation due to Kawakita [7] is adopted, because it is given in a simple algebraic form, which is easier to apply. Let p and ρ be, respectively, the powder medium pressure and the density at an arbitrary point in the die, and let a and b be coefficients determined by the medium. Then the relation between p and ρ is given by

$$\rho = \frac{1 + bp}{1 + (b - ab)p}\rho_0 \tag{11}$$

Eliminating c from equations (3) and (4), we obtain

$$p = \frac{v^2}{1/\rho_0 - 1/\rho} \tag{12}$$

Elimination of ρ from equations (11) and (12) yields $abp^2 - b\rho_0 v^2 p - \rho_0 v^2 = 0$. Putting $\alpha = ab$, $\beta = b\rho_0 v^2$, $\gamma = \rho_0 v^2$, we get equation

$$p = \frac{\beta + \sqrt{\beta^2 + 4\alpha\gamma}}{2\alpha} \tag{13}$$

From equation (4), we obtain

$$c = \frac{p}{\rho_0 v} = \frac{\beta + \sqrt{\beta^2 + 4\alpha\gamma}}{2\alpha\rho_0 v} \tag{14}$$

The time t required for the shock wave front to reach the point x is given by

$$t = \int_0^x \frac{1}{c} dx \tag{15}$$

By integrating equation (5) with respect to v using equations (10) and (14), we obtain

$$t = \frac{bMv_0}{2A}\left[\frac{\sqrt{v^2+k^2}}{v} - \frac{\sqrt{{v_0}^2+k^2}}{v_0} + \ln\left(\frac{v}{v_0}\right)\frac{\sqrt{v^2+k^2}-v-k}{\sqrt{v^2+k^2}+v-k}\,\frac{\sqrt{{v_0}^2+k^2}+v_0-k}{\sqrt{{v_0}^2+k^2}-v_0-k}\right], \tag{16}$$

where $k = \sqrt{\frac{4\alpha}{b\rho_0}}$. Thus, from equations (10), (11) and (13) the density distribution'in the medium after impaction at point x can be determined.

Finitely Long Metal Powder Medium in the Die

When the die length is finite, the shock wave is reflected at the other end, and moves back toward the punch. If the plug end $x = l_0$ is fixed, the medium at each point maintains constant density after the passing of the initial wave front because the pressure at points behind the reflected wave front is always lower than in front of it. Thus, the particle velocity becomes zero at any point behind the shock, which is reflected at the plug, and the pressure in the area remains constant, becoming a function of time alone, because the velocity of release adiabat becomes infinite due to neglecting elastic recovery. Let, as in the effectively infinite case, the pressure and the particle velocity in front of the wave front be $\bar{p}$, $\bar{\rho}$, $\bar{v}$ and p, ρ, v behind the wave front. Then, the equations of conservation of mass and momentum (1) and (2) become, respectively,

$$\rho(c - v) = \bar{\rho}(c - \bar{v}) \tag{17}$$

$$p - \bar{p} = \bar{\rho}(c - \bar{v})(v - \bar{v}) \tag{18}$$

Since $\bar{v} = 0$ in this case also, we have

$$\rho(c - v) = \bar{\rho} c \tag{19}$$

$$p - \bar{p} = \bar{\rho} c v \tag{20}$$

The pressure-density relation of Kawakita is given by

$$\bar{\rho} = \frac{1 + b\bar{p}}{1 + (b - ab)\bar{p}}\rho_0 \tag{21}$$

From equations (19), (20) and (21) we obtain

$$\bar{p} = \frac{\bar{\beta} + \sqrt{\bar{\beta}^2 + 4\bar{\alpha}\bar{\gamma}}}{2\bar{\alpha}} \tag{22}$$

where

$$\left.\begin{aligned} \bar{\alpha} &= b(\rho_0 - \rho + a\rho) \\ \bar{\beta} &= (\rho - \rho_0)(1 - bp) + b\rho(ap + \rho_0 v^2) \\ \bar{\gamma} &= \rho_0 p - \rho p + \rho_0 \rho v^2 \end{aligned}\right\} \tag{23}$$

When the wave front travels toward the plug end after reflection of the wave front at the punch surface, the particle velocity v remains always zero in front of the wave front. Thus,

$$\rho(c - v) = \bar{\rho} c \tag{24}$$

$$p - \bar{p} = \bar{\rho} c v \tag{25}$$

And α, β and γ in equation (13) are, respectively, given by

$$\left.\begin{aligned} \alpha &= b(\rho_0 - \bar{\rho} + a\bar{\rho}) \\ \beta &= (\bar{\rho} - \rho_0)(1 - b\bar{p}_0) + b\bar{\rho}(a\bar{p}_0 + \rho_0 v^2) \\ \gamma &= \rho_0 \bar{p}_0 - \bar{\rho}\bar{p} + \rho_0 \bar{\rho} v^2 \end{aligned}\right\} \tag{26}$$

Equations (5') and (6) remain valid for the reflected wave from the equation of momentum behind the wave front.

For the reflected wave moving toward the punch, powder medium mass is given by

$$m = m_0 + A \int_{t_i}^{t} \bar{\rho} c \, dt \quad (i = 1, 3, 5, \cdots) \tag{27}$$

Here ρ and p are known functions; $\bar{\rho}$, c, $\bar{p}$, v and m can be determined from equations (19), (20), (5'), (22) and (27), where t_i is the moment when the wave front is at the ends of the medium. For instance, t_0 is the moment of impaction, t_1 is the moment the wave front reaching the plug end for the first time, t_2 the moment the reflected wave reaching the punch surface for the first time, etc. The medium mass m in front of the wave front when the shock wave is propagating toward the plug end is given by

$$m = A \int_{t_i}^{t} \bar{\rho} c \, dt \quad (i = 0, 2, 4, 6, \cdots) \tag{28}$$

The pressure $\bar{p}$ is known, $\bar{p} = 0$ before the front wave reaches the plug end for the first time, and it is the pressure behind the reflected wave, ρ is also known; thus the unknown values ρ, c, v, p and m can be determined from equations (19), (20), (5'), (13) and (28). The necessary calculation can be done by use of the difference method.

Comparison of Experimental and Theoretical Results

Effectively Infinitely Long Powder Medium in the Die

Experimental and calculated data are shown in Fig. 2 where the solid line indicates experimental data and the dotted line indicates the calculated data. These were obtained for punch path u_p, shock wave front path u_f and powder particle path u_x for the situation where the punch of mass M = 785.781[g] was impacted against the effectively infinitely long powder medium in the die which had an initial density of ρ_i = 1.864[g/cm^3] at velocity v_0= 46[m/s]. As described in above sections, it is most likely that the travel of the reflection wave returning from the plug was absorbed before it could reach the section approximately 18[cm] from the end where the punch impacted the surface of the medium in the die. It can be clearly seen from the figure that with the passage of time a considerable variance occurs between the experimental and theoretical data, the experimental data being below the theoretical one. This is probably attributable to the friction between the powder and the die walls and the air pressure in the powder medium, for these factors cause a fall in the particle and wave front velocities. The frictional influence was especially appreciable and no displacement was noted in the element between the position x'= 40[cm] and plug although the particle velocity at

x'= 95[cm] was theoretically about one half of the impact velocity. However, the theoretical data [2] agrees with the experimental one [4] in that the powder particles move approximately as fast as the punch when the wave front arrives, and although it is not clear from this figure, the powder medium element gradually becomes longer, i.e., less dense, closer to the plug. Although there is a quantitative discrepancy with the theoretical data as noted, the experimental results agree well with the qualitative prediction the theory.

Finitely Long Powder Medium in the Die

Figures 3 (a), (b) and (c) show the experimental values of u_p, u_f and u_x in solid line and the calculated values of u_p, u_f and u_x in dotted line, which were obtained for the situation where the punch of mass M = 612.144[g] was impacted against a powder medium of length l_0= 5[cm], ρ_i=1.864[g/cm^3] at velocities 25, 29 and 34[m/s]. Figures 4 (a), (b) and (c) similarly show the data obtained when the punch was impacted against a 10[cm] long powder medium at velocities 24, 29 and 34[m/s]. The travel of the shock wave in the powder was also observed photographically with the aid of the high-speed camera. The results agree with the theory that powder particles in front of the wave front are at rest, and powder particles behind the wave front move approximately parallel with the punch path. After the wave front is reflected at the plug end, behind the wave front, the particles come to rest, and in front of the wave front, powder particles are still moving in the same direction and with the same velocity as the punch. It can also be noted from these figures that the compaction time decreases and the passage frequency of the wave front increases with increasing impact velocity and decreasing powder length.

The deformation of the medium can be seen more clearly than under other conditions and at least one single travel of the wave front can be observed. The experimental values of Fig. 3 and 4 are those measured at the center of line in the photograph.

The theoretical data is in excellent agreement with the experimental one, although the effect of air pressure may be more apparent at l_0= 5[cm] of Fig. 3. This result and the data obtained for the "infinitely" long powder medium suggest that the frictional influence decreases with decreasing sample length. The theoretical and experimental variance noted in the compaction of finitely long powder media might have been caused by the air pressure factor working in conjunction with the friction factor.

Figures 5 (a) and (b) show comparisons between the experimental values and the calculated values obtained for final mean density ρ_{mean} for situations where punches of three different masses 612.144, 459.108 and 306.072[g] were impacted against 5[cm] and 10[cm] long powder medium having initial densities of ρ_i=1.864[g/cm^3] at different velocities. In these experiments only steel plate was used and no acryl plates were used. As can be seen from Figs. 3 and 4, the experimental values are lower than the calculated values but both sets of data show a very similar tendency. Compact density increases with increasing punch mass and increasing initial velocity and decreasing powder medium length. Figure 6 shows a relation between mean green density and initial punch kinetic energy W_0 derived from the data of Figs. 5 (a) and (b). The final mean density is substantially independent of the punch mass both theoretically and experimentally when the initial kinetic energy is constant. This result can be confirmed theoretically from the fact that so far as calculations are possible, the initial kinetic energy is converted almost completely into powder medium strain energy and the compact attains an approximately uniform density.

Figure 7 gives the calculated and experimental values of mean density of the compact, which becomes approximately constant both in theory and experiment if the initial powder medium mass is constant, except that the compaction time and the amount of punch displacement undergo changes.

Conclusions

For the "effectively infinitely" long powder medium an analytical solution was obtained in which the front position x (expressed by the Lagrangean coordinate) was used as the variable, and for the finitely long medium a numerical solution was obtained. Also with the use of the Lagrangean coordinate, from the result for the infinitely long medium it has been confirmed that in all cases the element immediately behind the shock wave front lies where the pressure decrease occurs approximately as theoretically predicted and the compact of approximately uniform density is obtained by the high velocity compaction if the die wall friction is neglected. The behavior of the medium was filmed with the high-speed camera and the high velocity compaction process was thus investigated experimentally. Results obtained were compared with the theoretical results for the parameters used in the experiment.

As the shock wave propagates through the medium, in the "effectively infinite" case, particles just move at the punch velocity as the shock wave front reaches them, but in the finite length case, particles again come to rest when the reflected wave front arrives. These results agree with the theoretical prediction. In the "effectively infinite" case, the friction force increases with the advance of the front, and the compaction process data obtained differs considerably from the theoretical predictions. In the finitely long medium case, there is excellent agreement with the theoretical prediction possibly because of the high number of reflected passes of the shock wave between the punch and plug, so that the influence of die wall friction can be neglected. In this case a uniform density compact is obtained. Higher densities and uniform density distributions are obtained when the powder length is reduced and the mass and impact velocity of the punch are increased.

References

[1] Richitmyer, R.D, Morton.K.W.; Difference methods for initial value problems, Inter science Pub., 1957.
[2] Sano,Y.; Journal of the Japan Society of Powder and Powder Metallurgy 21-1, P. 1, 1974.
[3] Sano,Y. Sugita,T.; Journal of the Japan Society of Powder and Powder Metallurgy, 22-2, P.47, 1975.
[4] Sano,Y., Hgiwara,T. and Miyagi,K.; Journal of the Japan Society of Powder and Powder Metallurgy,.21-1, P.9, 1974.
[5] Sano,Y. and Miyagi,K.; The International Journal of Powder Metallurgy & Powder Technology, 20-2, P.115, 1984.
[6] Sano,Y.,Miyagi,K. and Hirose,T.;The International Journal of Powder Metallurgy & Powder Tech.,14-4, P.291,1978.
[7] Kawakita,T.; Journal of the Japan Society of Powder and Powder Metallurgy, 10-2, P.31, 1963.

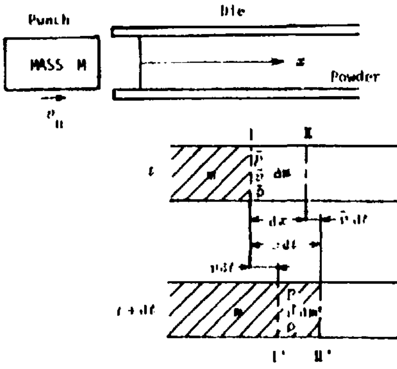


Fig. 1 Mathematical model

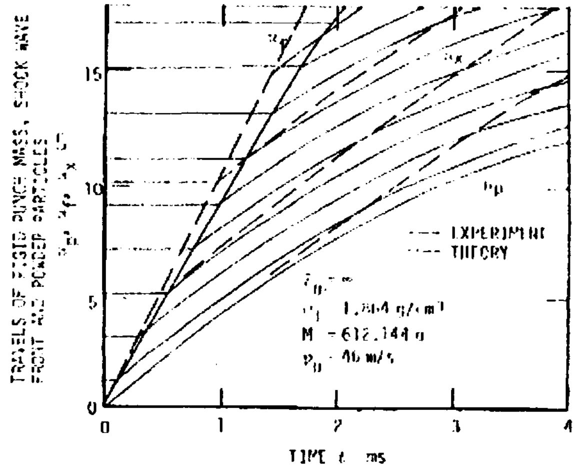
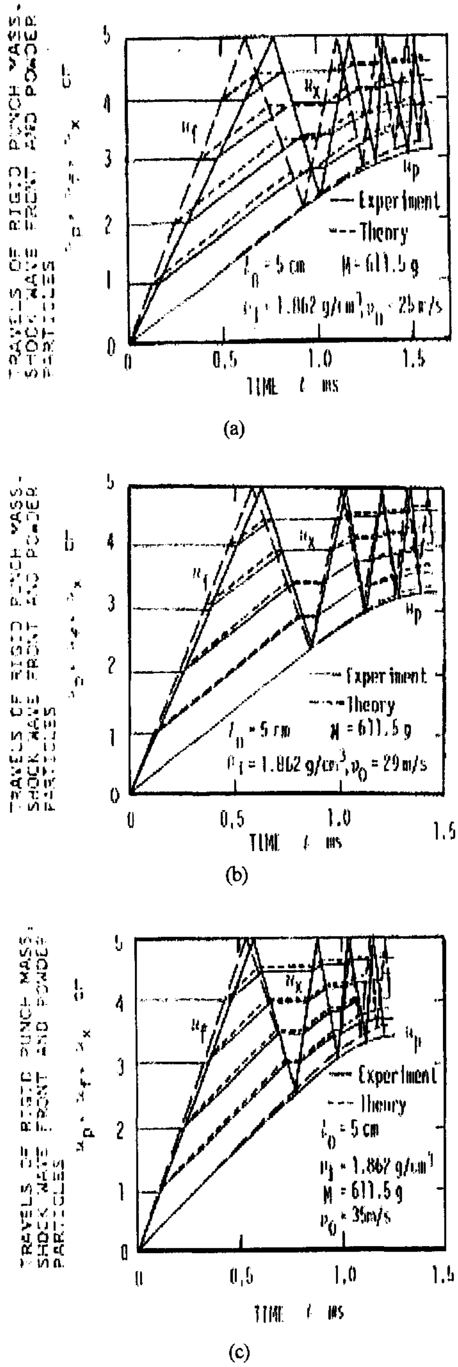


Fig .2 Time variations of the rigid punch mass, shock wave front and powder particles

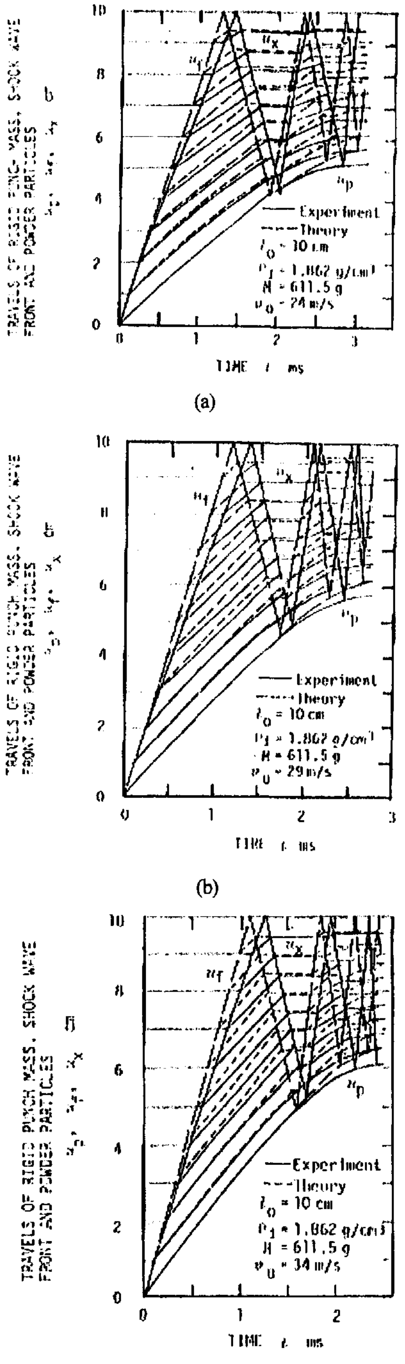


(a)

(b)

(c)

Fig. 3 Time variations of the rigid punch mass, shock wave front and powder particles

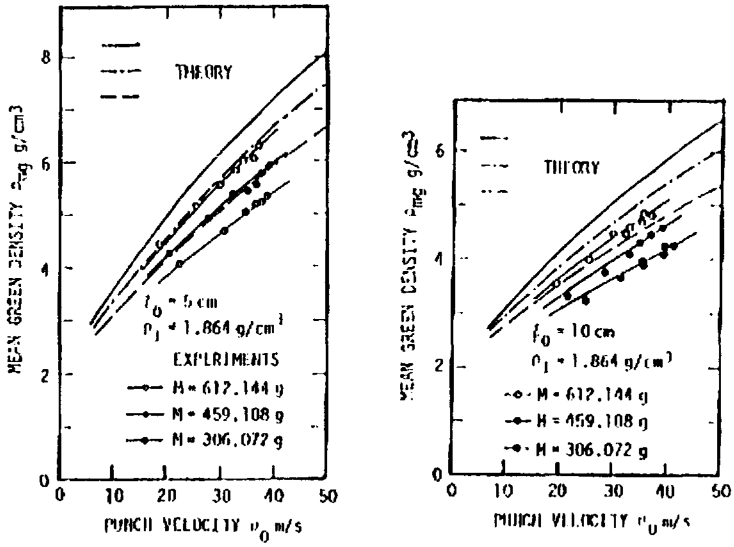


(a)

(b)

(c)

Fig. 4 Time variations of the rigid punch mass, shock wave front and powder particles



(a)

(b)

Fig. 5 Mean green density vs. punch velocity

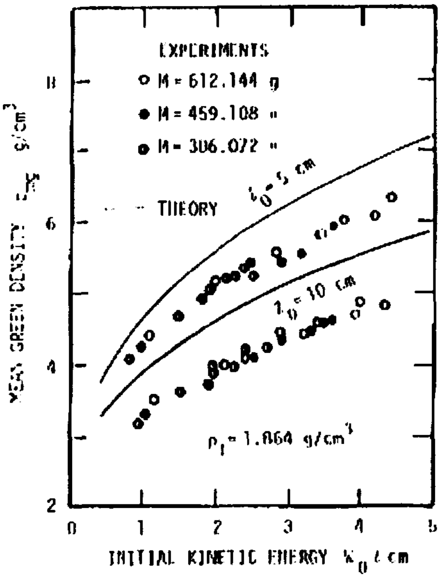


Fig. 6 Mean green density vs. initial kinetic energy

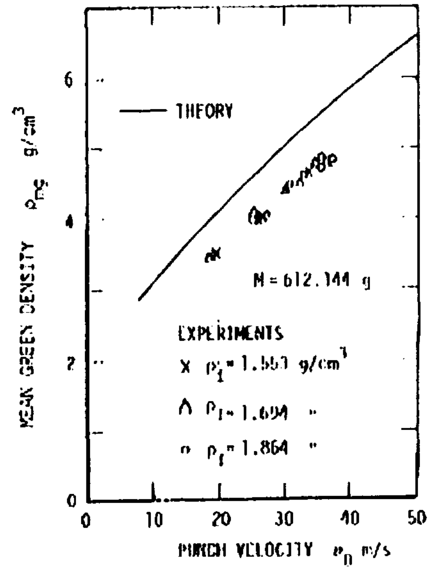


Fig. 7 Mean green density vs. punch velocity

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Automated Theorem Proving for Many-Sorted Free Description Theory Based on Logic Translation

Kazumi NAKAMATSU [1] and Atsuyuki SUZUKI [2]

[1] Himeji Institute of Technology, Shinzaike 1-1-12, HIMEJI 670-0092 JAPAN
`nakamatu@hept.himeji-tech.ac.jp`
[2] Shizuoka University, Johoku 3-5-1, HAMAMATSU 432-8011 JAPAN
`suzuki@cs.inf.shizuoka.ac.jp`

Abstract. In this paper, we provide an algorithm of automated theorem proving for a many-sorted free description theory with equality, and prove its completeness. We propose translation rules to translate the many-sorted free description theory with equality into first order many-sorted theory with no description. Based on the translation, we show that the automated theorem proving for the many-sorted free description theory can be implemented by an automated deduction method for first order(standard) many-sorted theory. We use an extended version of the RUE-NRF deduction method proposed by D.J.Diglicoli as the automated deduction method.

Keywords : free logic, automated theorem proving, many-sorted free description theory with equality, first order many-sorted theory, RUE-NRF deduction.

1 Introduction

Some languages for knowledge representation have been being studied in the field of artificial intelligence such as the field of knowledge representation and natural language understanding, eg. KRL(Knowledge representing Languages) [1], D-script proposed by Moore [6] etc. are those. If we consider to implement reasoning systems by those languages, which are implemented by programming languages such as PROLOG, it might be required to translate the languages to usual predicate logics so that they can be implemented in PROLOG, or might be required to consider automated deduction systems(theorem proving systems). It is known that KRL can be translated into a many-sorted free description theory with equality except for reflexive reasoning [5].

Free logics are logics that can deal with undefined objects(non-existing objects) [13, 15], and are indispensable for dealing with *description*. We give an algorithm of the automated theorem proving for this many-sorted free description theory with equality called FD_n in this paper, and show that the theorem prover is complete with respect to the translation from FD_n into a first order predicate logic. FD_n is an extension of Scott [13]. The automated theorem proving method proposed in this paper is based on the translation between free description theory and standard first order theory. Based on the translation, an automated deduction method for standard for the standard many-sorted theory is proposed to implement the many-sorted description theorem prover. The translation-based automated deduction method was originally proposed in Nakamatsu and Suzuki [7, 14] as theorem provers for modal logics and has been developed by Nakamatsu et al. [8, 9, 10, 11, 12] as the annotated semantics for some non-classical logics.

This paper is organized as follows. First, we show that any formula of FD_n can be translated into formulas of a standard many-sorted theory with equality called SE_n, and that the translation preserves provability, i.e., P is a theorem of FD_n iff the translation of P is a theorem of SE_n. Next, we describe the theorem prover for the many-sorted theory with equality and how to implement it. We use an automated deduction method that is an extension of the RUE-NRF deduction proposed by Diglicoli [3].

The reader is assumed to have basic knowledge of theorem proving and logic [2].

2 Formal Preliminaries

In this section, we define the many-sorted first order theory with equality SE_n and the many-sorted free description theory with equality FD_n.

2.1 Languages

We define many-sorted(n-sorted) languages L_s and L_f used for the thories SE_n and FD_n, respectively. The language L_s is defined as follows :

(1) variables $x_1, y_1, z_1, \ldots; x_2, y_2, z_2, \ldots; \ldots; x_n, y_n, z_n, \ldots$. (Subscripts indicate the sorts) ;

(2) function symbols $f_1, g_1, h_1, \ldots; f_2, g_2, h_2, \ldots; \ldots; f_n, g_n, h_n, \ldots$;

(3) 0-array function symbols are constants denoted by $a_1, b_1, c_1, \ldots; a_2, b_2, c_2, \ldots; \ldots; a_n, b_n, c_n, \ldots$;

(4) predicate symbols $P, Q, R, \ldots, E_1, \ldots, E_n, \ldots$;

(5) logical symbols $\sim$(negation), $\rightarrow$, $\forall$, $=$(equality) ;

(6) auxiliary symbols (,).

Other logical symbols, $\wedge, \vee, \equiv$ and existential quantifier $\exists$ are as abbreviations in the ordinary way.

The formation rules for terms and formulas are :

(1) any variables and 0-aray functions are terms ;

(2) if each $f_i (i = 1, \ldots, n)$ is an m-ary function symbol, then $f_i(t^1, \ldots, t^m)$ is a term, where $t^1, \ldots, t^m$ are any terms ;

(3) if t and s are terms of the same sort, then $t = s$ is a formula ;

(4) if $t^1, \ldots, t^m$ are terms of any sort and P is a predicate symbol, then $P(t^1, \ldots, t^m)$ is a formula ;

(5) if A and B are any formulas and each $x_i (i = 1, \ldots, n)$ is a variable of the i-th sort, then $\sim A$, $A \rightarrow B$, $\forall x_i A$ are formulas.

The language L_f is obtained by adding the following description symbol ι and formation rule to L_s :

(6) if each $x_i (i = 1, \ldots, n)$ is a variable of the i-th sort and A is a formula, then $\iota x_i A$ is a term.

We note that each $E_i (i = 1, \ldots, n)$ is a particular unary predicate symbol in L_s not occurring in L_f.

2.2 Syntax

We describe the axiom schemata and the inference rules of SE_n and FD_n.

Axiom of SE_n : let A, B are any formulas, each x_i $(i = 1, \ldots, n)$ is any variable of the i-th sort, and t,s are any terms of the same sort.

A1. A, if A is a tautology.

A2. $\forall x_i (A \rightarrow B) \rightarrow (\forall x_i A \rightarrow \forall x_i B)$.

A3. $t = t$.

A4. $t = s \wedge P(\ldots, t, \ldots) = P(\ldots, s, \ldots)$, where P is a predicate symbol.

A5. $t = s \wedge f_i(\ldots, t, \ldots) = f_i(\ldots, s, \ldots)$, where f_i is a function symbol of the i-th sort $(i = 1, \ldots, n)$.

Inference Rules of SE_n : let A, B are any formulas and each $x_i (i = 1, \ldots, n)$ is a variable of the i-th sort.

R1. If A and $A \rightarrow B$, then B (Modus Ponens). We write A to mean that A is provable.

R2. If $A \rightarrow B$ and a variable x_i of the i-th sort $(i = 1, \ldots, n)$ is not free in A, then $A \rightarrow \forall x_i B$.

The axiom schemata of FD_n can be obtained by adding the following axiom schemata, and the inference rules of FD_n are same as those of SE_n.

A6. $\forall y_i \exists x_i (x_i = y_i)$.

A7. $\forall y_i (y_i = \iota x_i A(x_i) \equiv \forall x_i (x_i = y_i \equiv A(x_i)))$.

A8. $\sim \exists (y_i = \iota x_i A(x_i)) \rightarrow \iota x_i A(x_i) = \iota x_i (x_i \neq x_i)$.

x_i and y_i $(i = 1, \ldots, n)$ are variables of the i-th sort.

2.3 Semantics

A model of SE_n is an ordered $n+1$-tuples $M =< D_1, \cdots, D_n, V >$, where each $D_i (i = 1, \ldots, n)$ is a non-empty set called a domain and V is a value assignment satisfying the following conditions :

(1) for each variable x_i, $V(x_i) \in D_i (i = 1, \ldots, n)$;

(2) for each constant (0-ary function symbol) c_i, $V(c_i) \in D_i (i = 1, \ldots, n)$;

(3) for each m-ary function symbol $f_i (m > 0)$, $V(f_i)$ is an unique mapping from $D_{j_1} \times \cdots \times D_{j_m}$ to D_i and satisfying the following condition.

$$V(f_i(t^1, \ldots, t^m)) = V(f_i)(V(t^1), \ldots, V(t^m)),$$

where each $j_k (k = 1, \ldots, m)$ is one of integers $1, \ldots, n$ and $i = 1, \ldots, n$:

(4) for each m-ary predicate symbol P, $V(P)$ is some set of ordered m-tuples, each of form $< \bar{u}^1_{i_1}, \ldots, \bar{u}^m_{i_m} >$, where each $\bar{u}^j_{i_j} \in D_{i_j} (j = 1, \ldots, m)$ and i_j is one of integers $1, \ldots, n$.

For any formula A in SE_n, the value assignment $V(A)$ is defined recursively :

(1) for each atomic formula $P(t^1, \ldots, t^m)$,

$$V(P(t^1, \ldots, t^m)) = 1 \quad \textbf{iff} \quad < V(t^1), \ldots, V(t^m) > \in V(P),$$
$$\text{otherwise} \quad V(P(t^1, \ldots, t^m)) = 0;$$

(2) for any terms s and t of the same sort,

$$V(s = t) = 1 \quad \textbf{iff} \quad V(s) = V(t), \quad \text{otherwise } V(s = t) = 0;$$

(3) for any formulas A and B,

$$V(\sim A) = 1 \quad \textbf{iff} \quad V(A) = 0, \quad \text{otherwise } V(\sim A) = 0,$$
$$V(A \to B) = 1 \quad \textbf{iff} \quad \text{if } V(A) = 1 \text{ then } V(B) = 1, \ \text{otherwise } V(A \to B) = 0;$$

(4) for any formula A and variable x_i of the i-th sort $(i = 1, \ldots, n)$,

$$V(\forall x_i A) = 1 \quad \textbf{iff} \quad \forall \bar{d}_i \in D_i, U(A) = 1,$$

where U is the value assignment such that $U(x_i) = \bar{d}_i$ and $U(y_i) = V(y_i)$ for distinct from $x_i (i = 1, \ldots, n)$.

[Definition 1]

A formula A in SE_n is valid **iff** for all assignment V in SE_n-model $V(A) = 1$.

A model of FD_n is an ordered $2n+1$-tuples $< G_1, \ldots, G_n, \hat{G}_1, \ldots, \hat{G}_n, U >$, where each G_i is the domain of properly existing individuals of the i-th sort $(i = 1, \ldots, n)$, each $\hat{G}_i$ is the domain of improperly existing individuals of the i-th sort ($\hat{G}_i$ is non-empty), and U is a value assignment satisfying the following conditions :

[1] for each variable x_i of the i-th sort, $U(x_i) \in (G_i \cup \hat{G}_i)(i = 1, \ldots, n)$:

[2] for each constant c_i of the i-th sort, $U(c_i) \in (G_i \cup \hat{G}_i)(i = 1, \ldots, n)$:

[3] for each m-ary function symbol $(m > 0)$ f_i, $U(f_i)$ is an unique mapping from $(G_{j_1} \cup \hat{G}_{j_1}) \times \cdots \times (G_{j_m} \cup \hat{G}_{j_m})$ to $(G_{j_i} \cup \hat{G}_{j_i})$ satisfying the following condition.

$$U(f_i(t^1, \ldots, t^m)) = U(f_i)(U(t^1), \ldots, U(t^m)),$$

where each $j_k (1 \leq k \leq m)$ is one of integers $1, \ldots, n$ and $i = 1, \ldots, n$:

[4] for each m-ary predicate symbol P, $U(\bar{P})$ is some set of ordered m-tuples, each of form $< \bar{W}^1_{i_1}, \ldots, \bar{W}^m_{i_m} >$, where each $\bar{W}^j_{i_j} \in (G_{i_j} \cup \hat{G}_{i_j} (j = 1, \ldots, m)$ and i_j is one of integers $1, \ldots, n$;

[5] for any formula A and variable x_i of the i-th sort $(i = 1, \ldots, n)$,

$$\begin{array}{rcll} U(\iota x_i A) & = & \bar{e}_i, & \text{if } e_i \text{ is the unique element of } G_i \text{ such that } U(A) = 1, \\ & & \bar{a}_i, & \text{if there is no such element, where } U(a_i) = \bar{a}_i \text{ and } \bar{a}_i \in \hat{G}_i. \end{array}$$

The assignments for formulas in FD_n are defined as well as those for in SE_n except for (4). The assignment $U(\forall x_i A)$ is defined by modifying the item (4) to the item (4') as follows :

(4') for any formula A and variable x_i of the i-th sort $(i = 1, \ldots, n)$,

$$U(\forall x_i A) = 1 \quad \textbf{iff} \quad \forall \bar{e}_i \in G_i, V(A) = 1,$$

where V is the value assignment such that $V(x_i) = \bar{e}_i$ and $V(y_i) = U(y_i)$ for distinct from $x_i (i = 1, \ldots, n)$.

[Definition 2]

A formula A in FD_n is valid **iff** for all assignment U in FD_n-model $U(A) = 1$.

Note : the system of FD_n can be regarded as an extension and a modification of the Scott's system [13].

3 Reducibility of FD_n to SE_n

Some methods to translate modal predicate logics to two-sorted first order logics have been proposed by Nakamatsu and Suzuki [7, 14]. In this section, we propose translation rules from FD_n into SE_n and show that the translation preserves provability between FD_n and SE_n.

3.1 *Translation Rule*

[Definition 3] (*-translation)
Given any term t and a formula A in FD_n, *-translation is defined as follows :

1. for any variable x_i of the i-th sort, $(x_i)* = x_i (i = 1, \ldots, n)$;
2. for any constant c_i of the i-th sort, $(c_i)* = c_i (i = 1, \ldots, n)$;
3. for any function symbol f_i of the i-th sort, $(f_i(t^1, \ldots, t^m))* = f_i(t^1*, \ldots, t^m*)$;
4. for a description $\iota x_i A(x_i) (i = 1, \ldots, n)$,

$$\begin{array}{rcl} (\iota x_i A(x_i))* & = & \iota x_i((E_i(x_i) \wedge \forall y_i(E_i(y_i) \to (A * (y_i) \equiv x_i = y_i))) \\ & & \vee(\sim (E_i(x_i) \wedge \forall y_i(E_i(y_i) \to (A * (y_i) \equiv x_i = y_i))) \wedge x_i = a_i)), \end{array}$$

where a_i is a particular constant in SE_n such that $U(a_i) = \bar{a}_i$ and $\bar{a}_i \in \hat{G}_i$;

5. for an atomic formula $P(t^1, \ldots, t^m)$,

(i) if none of $t^1, \ldots, t^m$ contains a term of the form $\iota x_i A$,

$$(P(t^1, \ldots, t^m))* = P(t^1, \ldots, t^m),$$

(ii) let $P(\iota x_{i_1} A_1(x_{i_1}), \ldots, \iota x_{i_m} A_m(x_{i_m}))$ stand for $P(t^1, \ldots, t^m)$, where $\iota x_{i_1} A_1(x_{i_1}), \ldots, \iota x_{i_m} A_m(x_{i_m})$ are all outermost descriptions occurring in $t^1, \ldots, t^m$ left to right, then

$$(P(t^1, \ldots, t^m))* =$$
$$\forall x_{i_1} \cdots \forall x_{i_m} (A_1 * (x_{i_1}) \wedge \cdots \wedge A_1 * (x_{i_m}) \rightarrow P(x_{i_1}, \ldots, x_{i_m})),$$

where each $i_j (1 \leq j \leq m)$ is one of $1, \ldots, n$;

6. let A and B be any formulas, $(\sim A)* = \sim A*$, $(A \rightarrow b)* = A* \rightarrow B*$, $(\forall x_i A)* = \forall x_i (E_i(x_i) \rightarrow A*)(i = 1, \ldots, n)$.

3.2 From FD_n to SE_n

Given a FD_n-model $M = < G_1, \ldots, G_n, \hat{G}_1, \ldots, \hat{G}_n, U >$, a SE_n-model $M* = < D_1, \ldots, D_n, V >$ can be defined by the *-translation. For each $i = 1, \ldots, n$:

- $D_i = G_i \cup \hat{G}_i$;
- for any variable x_i of the i-th sort, $V(x_i) = U(x_i)$;
- for a particular constant a_i of the i-th sort, $V(a_i) \in \hat{G}_i$;
- for the other constants c_i of the i-th sort, $V(c_i) = U(c_i)$;
- for any function symbol f_i of the i-th sort, $V(f_i) = U(f_i)$;
- for a particular predicate E_i not occurring in FD_n, $V(E_i) = G_i$;
- for the other predicate P, $V(P) = U(P)$.

Then we have the following lemma.
[Lemma 1] For any formula A in FD_n,

$$U(A) = 1 \text{ in } FD_n\text{-model } M \quad \textbf{iff} \quad V(A*) = 1 \text{ in } SE_n\text{-model } M*.$$

[Proof] We proceed by the induction on the length of the formula A.
[Case 1] In this case, A contains no description.
Basis. For an atomic formula $P(t^1, \ldots, t^m)$, If none of $t^1, \ldots, t^m$ contains a description,

$$(P(t^1, \ldots, t^m))* = P(t^1*, \ldots, t^m*) = P(t^1, \ldots, t^m).$$

Hence,

$$U(P(t^1, \ldots, t^m)) = 1 \quad \textbf{iff} \quad V(P(t^1, \ldots, t^m)) = 1 \quad \textbf{iff} \quad V((P(t^1, \ldots, t^m))*) = 1.$$

Induction Step. If the formula A has the form of $\sim B$ or $B \rightarrow C$, the result is clear.

$$U(\forall x_i B(x_i)) = 1 \quad \textbf{iff} \quad \text{for all } \bar{e}_i \in G_i, U'(B(x_i)) = 1,$$

where U' is the assignment such that $U'(x_i) = \bar{e}_i$ and $U'(y_i) = U(y_i)$ for distinct from x_i $(i = 1, \ldots, n)$.

$$V((\forall x_i B(x_i))*) = 1 \ \textbf{iff} \ V((\forall x_i(E_i(x_i) \to B*(x_i))) = 1$$
$$\textbf{iff} \ \text{ for all } \bar{e}_i \in (G_i \cup \hat{G}_i), \ \text{ if } \bar{e}_i \in G_i, \ \text{ then } V'(B*(x_i)) = 1,$$

where V' is an assignment such that $V'(x_i) = \bar{e}_i$ and $V'(y_i) = V(y_i)$ for distinct from $x_i (i = 1, \ldots, n)$. Thus,

$$\text{if } U'(B(x_i)) = V'(B*(x_i)), \text{ then } U(\forall x_i B(x_i)) = 1 \ \textbf{iff} \ V((\forall x_i B(x_i))*) = 1.$$

From the induction hypothesis, $U'(B(x_i)) = V'(B*(x_i))$. Hence,

$$U(\forall x_i B(x_i)) = 1 \ \textbf{iff} \ V((\forall x_i B(x_i))*) = 1.$$

[Case 2] In this case, A contains description.
Basis. For an atomic formula $P(t^1, \ldots, t^m)$, we consider only the case such that the j-th argument t^j contains a term of the form $\iota x_i B(x_i)$, where $B(x_i)$ contains no description for simplicity. To prove the general case is easy by induction.

$$\begin{aligned}&(P(t^1, \ldots, \iota x_i B(x_i), \ldots, t^m))* = \\ &\forall x_i((E_i(x_i) \land \forall y_i(E_i(y_i) \to (B*(y_i) \equiv x_i = y_i)) \lor \\ &(\sim (E_i(x_i) \land \forall y_i(E(y_i) \to (B*(y_i) \equiv x_i = y_i))) \land x_i = a_i) \land P(t^1, \ldots, x_i, \ldots, t^m)).\end{aligned}$$

$$U(P(t^1, \ldots, \iota x_i B(x_i), \ldots, t^m)) = 1$$
$$\textbf{iff}$$

if $\bar{e}_i$ is the unique element of G_i such that $U(B(x_i)) = 1$ then $U(\iota x_i B(x_i)) = U(x_i) = \bar{e}_i$ and $U(P(t^1, \ldots, x_i, \ldots, t^m)) = 1$, or if there is no such element then $U(\iota x_i B(x_i)) = U(x_i) = U(a_i) = \bar{a}_i \in \hat{G}_i$ and $U(P(t^1, \ldots, x_i, \ldots, t^m)) = 1$.

$$V((P(t^1, \ldots, \iota x_i B(x_i), \ldots, t^m))*) = 1$$
$$\textbf{iff}$$
$$V(\forall x_i((E_i(x_i) \land \forall y_i(E_i(y_i) \to (B*(y_i) \equiv x_i = y_i)) \lor$$
$$(\sim (E_i(x_i) \land \forall y_i(E(y_i) \to (B*(y_i) \equiv x_i = y_i))) \land x_i = a_i) \land P(t^1, \ldots, x_i, \ldots, t^m))) = 1$$
$$\textbf{iff}$$

if $\bar{e}_i$ is the unique element of G_i such that $V(B*(y_i)) = 1$ and $V(x_i) = \bar{e}_i$, or there is no such element and $V(x_i) = V(a_i) \in \hat{G}_i$, then $V(P(t^1, \ldots, x_i, \ldots, t^m)) = 1$.

Since $B(x_i)$ contains no description, from **[Case 1]** and by induction,

$$U(B(x_i)) = 1 \ \textbf{iff} \ V(B*(y_i)) = 1.$$

Hence,

$$U(P(t^1, \ldots, \iota x_i B(x_i), \ldots, t^m)) = 1 \ \textbf{iff} \ V((P(t^1, \ldots, \iota x_i B(x_i), \ldots, t^m))*) = 1.$$

In **[Case 2]**, for $i = 1, \ldots, n$, Induction Steps are same as **[Case 1]**.

Q.E.D.

Given an SE_n-model $M* =< D_1, \cdots, D_n, V >$, a FD_n-model $M =< G_1, \cdots, G_n, \hat{G}_1, \cdots, \hat{G}_n, U >$ can be defined. For each $i = 1, \ldots, n$:

- $G_i = V(E_i)$, $\hat{G}_i = D_i \setminus G_i$;

- for a particular constant a_i of the i-th sort, $U(a_i) = V(a_i) = \bar{a}_i \in \hat{G}_i$, therefore each $\hat{G}_i$ is non-empty ;
- for the other constants c_i of the i-th sort, $U(c_i) = V(c_i)$;
- for any variable x_i of the i-th sort, $U(x_i) = V(x_i)$;
- for any function symbol f_i of the i-th sort, $U(f_i) = V(f_i)$;
- for any predicate P, $V(P) = U(P)$.

Thus, we have another lemma and it can be proved similarly to **[Lemma 1]**.
[Lemma 2] For any formula $A*$ of $*$-translation of A,

$$V(A*) = 1 \text{ in } SE_n - \text{model } M* \quad \textbf{iff} \quad U(A) = 1 \text{ in } FD_n - \text{model } M.$$

Considering the completeness of the *-translation, we have the following theorem.
[Theorem 1] For any formula A in FD_n,

$$A \text{ is provable in } FD_n \quad \textbf{iff} \quad A* \text{ is provable in } SE_n.$$

[Proof] We assume the completeness for FD_n and SE_n that can be proved by the similar way of Scott [13] and Wang [16].
Proof of **if**-part. Suppose A is provable in FD_n, we have that A is valid in FD_n. Moreover, A is valid in FD_n **iff** $U(A) = 1$ in every FD_n-model $M =< G_1, \ldots, G_n, \hat{G}_1, \ldots, \hat{G}_n, U >$. Then, from **[Lemma 1]**, $V(A*) = 1$ in the corresponding SE_n-model $M* =< D_1, \ldots, D_n, V >$. Hence, $A*$ is valid and provable in SE_n.
Proof of **only if**-part. Suppose $A*$ is provable in SE_n, we have that $A*$ is valid in SE_n. Moreover, $A*$ is valid in SE_n **iff** $V(A*) = 1$ in every SD_n-model $M* =< D_1, \ldots, D_n, V >$. Then, from **[Lemma 2]**, $U(A) = 1$ in the corresponding FD_n-model $M =< G_1, \ldots, G_n, \hat{G}_1, \ldots, \hat{G}_n, U >$. Hence, A is valid and provable in FD_n.

Q.E.D.

4 Automated Deduction for FD_n

In this section, we provide an algorithm for theorem proving of the many-sorted free description theory FD_n. We use the RUE and NRF by Diglicoli [3] that can deal with equality to construct the theorem prover for FD_n, which is based on the *-translation.

4.1 Algorithm for Theorem Proving

We refer to Chang[2] and Diglicoli[3] through the following algorithm.

[Step 1] For a given formula A of FD_n, translate $\sim A$ into $\sim A*$. Furthermore, rename bound variables, if necessary.

[Step 2] Transform the formula produced in the previous step into a Prenex Normal Form.

[Step 3] Skolemize the Prenex Normal Form $Q_1 x^1_{i_1} \ldots Q_m x^m_{i_m} M(x^1_{i_1}, \ldots, x^m_{i_m})$ in the previous step, where $Q_k (1 \leq k \leq m)$ is a quantifier, M is a matrix, and each $i_j (1 \leq j \leq m)$ is one of integers $1, \ldots, n$.

[Step 4] Construct the set of clauses from the matrix M.

[Step 5] The RUE-NRF deduction for many-sorted theory.

4.2 Completeness of Theorem Proving

We show that for any formula F is a theorem in FD_n iff there is a deduction of the empty clause $\Box$ from the set of clauses constructed in **[Step 4]**.
[Definition 4] (unsatisfiability)
If a formula F in SE_n is evaluated to 1 in a model $M =< D_1, \ldots, D_n, V >$, we say that M satisfies the formula F. A formula F is unsatisfiable(inconsistent) **iff** there exists no model that satisfies F.
[Definition 5] (E-satisfiability)
A formula F is called E-satisfiable **iff** there is a model that satisfies the formula F and the equality axioms that are stated below. Otherwise, F is called E-unsatisfiable.
[Equality Axiom]

$$1.x_i = x_i, \quad 2.x_i = y_i \rightarrow y_i = x_i, \quad 3.x_i = y_i \wedge y_i = z_i \rightarrow x_i = z_i,$$
$$4.x_i = y_i \rightarrow f_j(\ldots, x_i, \ldots) = f_j(\ldots, y_i, \ldots),$$
$$5.x_i = y_i \rightarrow P(\ldots, x_i, \ldots) = P(\ldots, y_i, \ldots).$$

Since a model of SE_n satisfies the equality axioms, any formula F in SE_n is unsatisfiable iff the formula F is E-unsatisfiable. Since it is straightforward that a formula F in FD_n is a theorem iff $\sim F*$ in SE_n is E-unsatisfiable from **[Theorem 1]**, we show that

(i) $\sim F*$ is E-unsatisfiable **iff** the clause set S constructed from $\sim F*$ is E-unsatisfiable.

(ii) the clause set S is E-unsatisfiable **iff** an RUE-NRF deduction of the empty clause $\Box$ from the clause set S. This completeness can be proved similarly to Diglicoli[3], because our theorem prover is an extension of the RUE-NRF to many-sorted theory.

It is obvious that **[Step 1]** and **[Step 2]** preserve E-unsatisfiablity. Thus, we show it for **[Step 3]** and **[Step 4]**.
[Theorem 2] Let S be a set of clauses that represents the standard form of the formula F in SE_n. Then,

the formula F is E-unsatisfiable **iff** the set S is E-unsatisfiable.

[Proof] It can be proved similarly to the proof of *Theorem 6* in Suzuki and Nakamatsu [14]. We have shown this theorem on two-sorted first order theory.

Q.E.D.

Here we give an outline of the RUE-NRF for many-sorted. We consider **4** and **5** of **[Equality Axiom]**. From **4**, we can deduce

$$f_i(s) \neq f_i(t) \rightarrow s \neq t (i = 1, \ldots, n), \tag{1}$$

where s and t are terms that are the same sort. From (1), we can infer $s \neq t$. This rule of inference is called *Negative Reflexive Function Rule* (NRF for short). Next from **5**, we can deduce $P(s)$ and $\sim P(t)$. This rule of inference is called *Resolution by Unification and Equality* (RUE for short). In order to give rigorous definitions of these rules later, we introduce some definitions of terminologies in theorem proving.
[Definition 6] (disagreement set)
Given two terms s, t of the same sort, a disagreement set is defined as :

- if s, t are identical, the empty set is the sole disagreement set ;
- if s, t are not identical, the set of one element the pair (s, t) is the origin disagreement set of s, t ;
- if s, t have the form $s = f_i(s^1, \ldots, s^m)$, $t = f_i(t^1, \ldots, t^m)$, respectively, then the set of pairs of the corresponding arguments that are not identical is the topmost disagreement set of s, t ;
- if D is a disagreement set s, t, then D' obtained by replacing any member of D by the elements of its own disagreement set is also a disagreement set of s, t.

A disagreement set of a pair of complementary literals $P(s^1, \ldots, s^m)$ and $\sim P(t^1, \ldots, t^m)$ is the union : $D = \cup_{j=1}^{m} D_j$, where D_j is a disagreement set of (s^j, t^j).

MGU(most general unifier) is used in standard resolution, on the other hand, MGPU(most general partial unifier) is used in the RUE-NRF. MGPU is the substitution used when two terms s, t are not completely unifiable.

[Definition 7] (difference set)
Let W be a non-empty set of expressions. The first *difference set* is obtained by locating the first position (counting from the left) at which not all the expressions in W have exactly the same symbol and then extracting from each expression the subexpression that begins with the symbol occupying that position. If we resume the comparison in each expression of W at the first position after the subexpression used to define the first difference set, find the next point of disagreement and again extract the corresponding subexpressions that comprise the second difference set of W. If the elements of W are not identical, k difference sets : $d^1, d^2, \ldots, d^k (k \geq 1)$ are constructed in this fashion.

Then, MGPU Algorithm is defined to compute a MGPU substitution for W.

[Definition 8] (MGPU, MGPU Algorithm)
MGPU Algorithm

Let W be a set of expressions and each d^j is a difference set defined in **[Definition 7]**.

(1) Set $j = 1$, $\sigma = \{\}$, where σ is a substitution.

(2) Find d^j for W as previously described, if it does not exist, terminate with σ a MGPU of the set W.

(3) If d^j contains a variable v_i of the i-th sort as members and a term t_i of the i-th sort that does not contain v_i, then let $\sigma = \sigma \cup \{t_i/v_i\}$ and $W = W\{t_i/v_i\}$. Go to (2).

(4) If d^j does not contain the above variable, then let $j = j + 1$ and go to (2).

The RUE Rule of Inference

Given clauses $A \vee P(s^1, \ldots, s^m)$, $B \vee \sim P(t^1, \ldots, t^m)$ and a substitution σ, the *RUE resolvent* of $\sigma(A \vee P(s^1, \ldots, s^m))$ and $\sigma(B \vee \sim P(t^1, \ldots, t^m))$ is $\sigma A \vee \sigma B \vee D$, where each s^j and $t^j (1 \leq j \leq m)$ are terms that of the same sort, and D is the disjunction of the equalities specified by a disagreement set of the complimentary literals σP and $\sigma(\sim P)$. σ may be the null substitution, the MGPU of $\{ P(s^1, \ldots, s^m), P(t^1, \ldots, t^m) \}$ or any substitution.

The NRF Rule of Inference

Given a clause $A \vee s \neq t$ and a substitution σ, the *NRF resolvent* of $\sigma A \vee \sigma s \neq \sigma t$ is $\sigma A \vee D$, where D is the disjunction of the inequalities specified by the disagreement set of $(\sigma s, \sigma t)$.

[Definition 9] (RUE-NRF deduction)

Given a set of clauses S, an *RUE-NRF deduction* of C from S is a finite sequence $C^1, C^2, \ldots, C^k$ such that each $C^j (1 \leq j \leq k)$ is either a clause in S or an RUE-NRF resolvent of the clauses preceding C^j and where $C^k = C$.

[Theorem 3] A set of clauses S is E-unsatisfiable **iff** there is an RUE-NRF deduction of the empty clause $\Box$ from S.

[Proof] Refer to Diglicoli[3].

4.3 Example for Theorem Proving

We show that the following formula is a theorem of FD_n as an example of the theorem proving.

$$\forall y(y = \iota x A(x) \to \forall x(x = y \equiv A(x)), \tag{2}$$

where x, y are variables of the i-th sort $(i = 1, \ldots, n)$. Let us describe the process of the RUE-NRF deduction of the formula (2).

*-translation of the formula (2),

$$\forall y(E(y) \to (\forall x P(x,y) \to \forall x Q(x,y))), \tag{3}$$

where

$$P(x,y) \equiv E(x) \to ((E(x) \to \forall z Q(z,x)) \vee (\sim (E(x) \wedge \forall z Q(z,x)) \wedge x = a) \to y = x),$$
$$Q(x,y) \equiv E(x) \to (x = y \equiv A*(x)), \quad E(x) \equiv E_i(x),$$

and z is a variable of the i-th sort. The negation of the formula (3) :

$$\exists y(E(y) \wedge P(x,y) \wedge \exists x \sim Q(x,y)).$$

Skolemization :

$$E(c) \wedge P(x,c) \wedge \sim Q(d,c),$$

where c, d are Skolem functions, and

$$\begin{aligned} P(x,c) \equiv\ & (c = x \vee \sim E(x) \vee \sim Q(f(x),x)) \wedge (c = x \vee x \neq a) \wedge \\ & (\sim E(x) \vee Q(z,x) \vee x \neq a) \wedge (\sim E(x) \vee Q(z,x) \vee x \neq a). \end{aligned}$$

Then, the set

$$\begin{aligned} S = \{ \quad & E(c), && (4) \\ & c = x \vee \sim E(x) \vee \sim Q(f(x),x), && (5) \\ & c = x \vee x \neq a, && (6) \\ & \sim E(x) \vee Q(z,x) \vee x \neq a, && (7) \\ & \sim Q(d,c) \quad \} && (8) \end{aligned}$$

of clauses is produced and by the RUE-NRF deduction,

$$\begin{aligned} & Q(z,c) \vee c \neq a, && \text{by RUE, from (4) and (7),} && (9) \\ & c \neq a, && \text{by RUE, from (8) and (9),} && (10) \\ & a \neq a, && \text{by RUE, from (6) and (10),} && (11) \\ & \Box \text{ (the empty clause),} && \text{by NRF, from (11).} && \end{aligned}$$

5 Conclusion

In this paper, we have given an algorithm for automated theorem proving for the many-sorted free description theory FD_n with equality. In the algorithm, the free description theory FD_n is translated into the standard many-sorted theory SE_n without description, and the translation is somewhat complicated. Thus, we need some strategies to improve the efficiency for the RUE-NRF deduction.

Considering the efficiency of theorem proving for many-sorted logic, to translate many-sorted logics into standard ones may not be adequate. Therefore, we give an automated theorem proving method for the many-sorted theory SE_n as it is.

References

[1] Bobrow,D.G. and Winograd,T. : An Overview of KRL, a Knowledge Representation Language. Cognitive Science **1**, No.1, (1977)

[2] Chang,C.L. and Lee, R.E. : Symbolic Logic and Mechanical Theorem Proving. Academic Press (1973)

[3] Diglicoli,V.J. : Automatic Deduction and Equality. Proc. of the 1979 Annual Conf. ACM. (1979) 240–250

[4] Grandy,R.E. : On the Relation between Free Description Theories and Standard Quantification Theory. NDFJL, **17** (1976)

[5] Hayes,P.J. : The Logic of Frames. in Frame Conceptions and Text Understanding. Walter de Gruyter (1980)

[6] Moore,R.C. : D-script : A Computational Theory of Descriptions. IEEE Trans. Computer **C-25** No.4 (1976)

[7] Nakamatsu,K. and Suzuki,A. : Automatic Theorem Proving for Modal Predicate Logic. Trans. IECE Japan, **E67** (1984) 203–210

[8] Nakamatsu,K. and Suzuki,A. : Annotated Semantics for Default Reasoning. Proc. PRICAI'94, (1994) 180–186

[9] Nakamatsu,K. and Suzuki,A. : A Non-monotonic ATMS Based on Annotated Logic Programs with Strong Negation. in Agents and Multi-Agent Systems. LNAI **1441** (1998) 79–93

[10] Nakamatsu,K., Abe,J.M. and Suzuki,A. : A Defeasible Deontic Reasoning System Based on Annotated Logic Programming. Proc. 4th Int'l Conf. Computing Anticipatory Systems. AIP Conf. Proc. **573** (2000) 609–620

[11] Nakamatsu,K., Abe,J.M. and Suzuki,A. : Annotated Semantics for Defeasible Deontic Reasoning. Rough Sets and Current Trends in Computing, LNAI **2005** (2001) 470–478

[12] Nakamatsu,K. : On the Relation Between Vector Annotated Logic Programs and Defeasible Theories. Logic and Logical Philosophy, **8** (2002) 181–205

[13] Scott,D. : Existence and Description in Formal Logic. in Bertrand Russell Philosopher of the Century(Schoenman,R. ed.). George Allen and Unwin Co. (1967)

[14] Suzuki,A. and Nakamatsu,K. : A Mechanical Theorem Proving of First-Order Modal Logic(S5). Trans. IECE Japan, **E65** (1982) 730–736

[15] van Frassen, B.C. and Lambert K. : On Free Description Theory. Zeitschrift für Mathematichen Logik und Grundlangen der Mathematik, **13** (1967)

[16] Wang,H. : Logic of Many-Sorted Theories. Journal of symbolic Logic. **17** (1952).

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Annotated Logic and Negation as Failure

Kazumi NAKAMATSU [1] and Atsuyuki SUZUKI [2]

[1] Himeji Institute of Technology, Shinzaike 1-1-12, HIMEJI 670-0092 JAPAN
`nakamatu@hept.himeji-tech.ac.jp`
[2] Shizuoka University, Johoku 3-5-1, HAMAMATSU 432-8011 JAPAN
`suzuki@cs.inf.shizuoka.ac.jp`

Abstract. In this paper we provide the declarative semantics for a derivation rule of negative information, Clark's Negation as Failure(NF for short), called annotated completion. The annotated semantics is based on a 4-valued annotated logic. We show that it can deal with inconsistency with taking a simple example and prove the soundness and completeness theorems with respect to the annotated semantics.
Keywords : annotated logic, inconsistency, declarative semantics, Negation as Failure.

1 Introduction

Some nonmonotonic reasonings such as default reasoning are utilized and the treatment of inconsistency has become more important in AI fields. For example, default reasoning for belief revision techniques is used in knowledge representation systems and the treatment of contradiction between agents in a multi-agent system is one of hot topics. Considering more intelligent knowledge systems, they would require more than two kinds of nonmonotonic reasoning as well as the capability to deal with inconsistency such as NOGOOD in a nonmonotonic ATMS. However, it is difficult to deal with these nonmonotonic reasonings and inconsistency uniformly, since they have different semantics. Thus, we try to uniformly represent the declarative semantics for such nonmonotonic reasonings and inconsistency based on annotated logic. We have already characterized some nonmonotonic reasonings with inconsistency by annotated logic programs [8, 9, 10, 11, 12, 13]. In this paper, we characterize the Negation as Failure by annotated semantics called annotated completion and prove its completeness and soundness.

Annotated logics are a family of paraconsistent logics that were proposed initially by Subrahmanian [14] and da Costa [3]. They have been applied to develop the declarative semantics for inheritance networks and object-oriented databases [15].

The derivation rules of negative information in logic programming systems and knowledge bases, Clark's Negation as Failure(NF for short) [2] is a nonmonotonic reasoning. Many researchers have given some logical interpretations to these rules. Fitting[4] and Kunen[6] provided an explanation of logic program with negation based on some logical models. Balbiani[1] and Gabbay[5] showed that SLDNF-provability has a modal meaning and it can be expressed in a modal logic. However, their semantics for these rules cannot deal with inconsistency such as contradiction between agents in multi-agent systems. Therefore, we give different declarative semantics for NF based on annotated logic to deal with inconsistency. With respect to these semantics, we show the soundness and completeness of NF.

This paper is organized as follows. First, Clark's Negation as Failure is introduced. Next, the annotated semantics called annotated completion is defined. Last, with respect to the annotated semantics, the soundness and completeness of NF are proved.

2 NF and Annotated Logic

NF rule is an inference rule in logic programming which gives a false value to a ground atom if the logic program cannot give a proof of that ground atom. If A is a ground atom NF rule can be interpreted informally as follows :

the goal $\neg A$ succeeds if A finitely fails,
the goal $\neg A$ finitely fails if A succeeds.

We give the semantics for the NF by means of the annotated completion for logic program, which differs from Clark's completion [2] where the underlying logic is annotated logic rather than classical logic. An annotated logic AL is a paraconsitent and multi-valued logic based on a complete lattice of truth values. However, AL provides a simple formalism that is closer to classical logic than ordinary multi-valued logic. For example, each atom in AL has a truth value as an annotation and the formula itself can be interpreted in classical logic fashion. Generally a goal(ground atom) A either succeeds, finitely fails or loops in logic programming. We consider the following complete lattice $\mathcal{T}$ in Figure 1 of truth values $\{\ l,\ f,\ s,\ \top\ \}$. The language of AL is similar to that of an ordinary logic. The only syntactic difference is that atomic formulas in AL are constructed from those in the ordinary logic by appending to them annotations that are drawn from the lattice $\mathcal{T}$. If A is a ground atom, then $(A : \mu)$ is called an *annotated atom*, where $\mu \in \mathcal{T}$. μ is called an *annotation* of A. AL has two kinds of negations : an *epistemic* negation $(\neg)$, a unary function from an annotation to an annotation such that $\neg(l) = l$, $\neg(f) = s$, $\neg(s) = f$, $\neg(\top) = \top$.

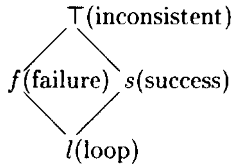


Figure 1: The Lattice $\mathcal{T}$ of Truth Values

We now adddress the semantics for AL. Let I be an interpretation and S_A be a set of ground atoms. An interpretation I of AL over $\mathcal{T}$ may be considered to be a mapping $I : S_A \rightarrow T$. Usually I is denoted by a set $\{\ (p : \sqcup\mu_i) | I \models (p : \mu_1) \wedge \cdots \wedge (p : \mu_n)\}$, where $\sqcup\mu_i$ is a least upper bound of $\{\mu_1, \cdots, \mu_n\}$.

[Definition 1] (satisfaction)
An interpretation I is said to satisfy

(1) a formula F **iff** I satisfies every closed instance of F ;

(2) a ground annotated atom $(A : \mu)$ **iff** $I(A) \geq \mu$;

(3) a formula $\sim F$ **iff** I does not satisfy F.

The satisfaction of the other formulas are the same as the ordinary logic. The details of the syntax and semantics for AL can be found in [3].

The reason motivating our choice of AL to study the problem of the semantics for NF are explained. An annotated atom $(A:\mu)$ can represent the derivation of the atom A by its annotation μ as follows :

$(A:l)$ the ground atom A is known to loop
i.e. A is known to neither succeed nor finitely fail ;

$(A:f)$ the ground atom A is known to finitely fail ;

$(A:s)$ the ground atom A is known to succeed ;

$(A:\top)$ the ground atom A is know to both succeed and finitely fail.

Usually the annotation $\top$ does not appear to characterize ordinary logic programs with NF. However, it is necessary for expressing inconsistency in such cases where multi-agent systems include contradiction between their agents.

We have the following equivalences based on the property of epistemic negation.

$$\neg(A:s) \equiv (A:\neg(s)) \equiv (A:f), \tag{1}$$
$$\neg(A:f) \equiv (A:\neg(f)) \equiv (A:s). \tag{2}$$

We can interpret the formulas (1) and (2) informally as :

(1) "a literal $\neg A$ succeeds" is equivalent to "the literal A finitely fails" ;

(2) "a literal $\neg A$ finitely fails" is equivalent to "the literal A succeeds".

The above (1) and (2) express the meaning of NF informally. Therefore, based on the idea, we construct the annotated semantics for NF in the following section.

3 Annotated Completion

The most widely accepted declarative semantics for NF uses the "completed database" introduced by Clark[2]. These formulas are usually called the *completion* or *Clark's completion* of a program P and denoted by $Comp(P)$. The aim of the completion is to logically characterize the SLD-finite failure set of the program P.

We propose the annotated completion formula and prove the soundness and completeness theorems for NF with respect to the annotated completion. First, we review about Clark's completion. The idea of Clark's completion is to consider that a predicate is totally defined by clauses whose heads have the same symbol as the predicate. This property is syntactically denoted in $Comp(P)$ as an equivalence between a predicate symbol and the disjunction of clause bodies. In the general case of $Comp(P)$, each clause

$$L_1, \cdots, L_m \rightarrow R(x_1, \cdots, x_n)$$

in the program P in which the predicate symbol R appears in the head is taken and rewritten in a general form :

$$\exists y_1 \cdots \exists y_k (x_1 = t_1 \wedge \cdots \wedge x_n = t_n \wedge L_1 \wedge \cdots \wedge L_m) \rightarrow R(x_1, \cdots, x_n),$$

where $x_1, \cdots, x_n$ are new variables (i.e., not already occurring in any of these clauses) and $y_1, \cdots, y_m$ are variables of the original clause. If the general forms of all these clauses are

$$E_1 \rightarrow R(x_1, \cdots, x_n), \cdots, E_j \rightarrow R(x_1, \cdots . x_n),$$

then the completed definition of R is

$$E_1 \vee \cdots \vee E_j \leftrightarrow R(x_1, \cdots, x_n).$$

The completion $Comp(P)$ of the predicate R is defined to be the set of completed definitions of each predicate symbol in P together with the equality and freeness axioms called CET(Clark's Equational Theory) [7]. We define an annotated completion formula $AComp(P)$ and prove that NF is sound and complete for the annotated completion $AComp(P)$. Let P be a program and

$$\forall x_1 \cdots \forall x_n (E_1 \vee \cdots \vee E_M \leftrightarrow R(x_1, \cdots, x_n))$$

be the completed definition of R in P. This formula is logically equivalent to the conjunction of formulas

$$\forall x_1 \cdots \forall x_n (E_1 \vee \cdots \vee E_M \rightarrow R(x_1, \cdots, x_n)),$$
$$\forall x_1 \cdots \forall x_n (\neg E_1 \wedge \cdots \wedge \neg E_M \rightarrow \neg R(x_1, \cdots, x_n)).$$

Each $E_i (1 \leq i \leq M)$ is of a form,

$$\exists y_1 \cdots \exists y_k (x_1 = t_1 \wedge \cdots \wedge x_n = t_n \wedge L_1 \wedge \cdots \wedge L_m)$$

and each negation $\neg E_i (1 \leq i \leq M)$ is of a form

$$\forall y_1 \cdots \forall y_k (x_1 = t_1 \wedge \cdots \wedge x_n = t_n \rightarrow \neg L_1 \vee \cdots \vee \neg L_m),$$

where the original clause is

$$L_1, \cdots, L_m \rightarrow R(t_1, \cdots, t_n).$$

Note : We assume the axiomatic system of AL including CET and the interpretation of equality is given as usual.

Let us replace each literal $L_i (1 \leq i \leq m)$ and $R(t_1, \cdots, t_n)$ by the corresponding annotated literal $(L_i : s)$ $(1 \leq i \leq m)$ and $(R(t_1, \cdots, t_n) : s)$ respectively.

[Definition 2] (*Annotated Completion* $AComp(P)$)

We obtain the following formulas that constitute the annotated completed definition of a predicate symbol R :

$$\forall x_1 \cdots \forall x_n (E_1^+ \vee \cdots \vee E_M^+ \rightarrow R(x_1, \cdots, x_n) : s), \tag{3}$$
$$\forall x_1 \cdots \forall x_n (E_1^- \wedge \cdots \wedge E_M^- \rightarrow R(x_1, \cdots, x_n) : f). \tag{4}$$

where each E_i^+ $(1 \leq i \leq M)$ is of a form,

$$\exists y_1 \cdots \exists y_k (x_1 = t_1 \wedge \cdots \wedge x_n = t_n \wedge (L_1 : s) \wedge \cdots \wedge (L_m : s)),$$

and each E_i^- $(1 \leq i \leq M)$ is of a form,

$$\forall y_1 \cdots \forall y_k (x_1 = t_1 \wedge \cdots \wedge x_n = t_n \rightarrow (L_1 : f) \vee \cdots \vee (L_m : f)).$$

We note P_1^+ the set of positive completed definition (3), P_1^- the set of negative completed definition (4) of the predicate R and define

$$ACom p(P) = P_1^+ \cup P_1^- .$$

In $AComp(P)$, the annotated atom $(A : s)$ that has s as its annotation should be interpreted as "A succeeds in P", that is to say, "there is the SLD-refutation of A in the program P" and the annotated atom $(A : f)$ that has f as its annotation should be interpreted as "A fails finitely in the program P", that is to say, "there is the SLD-finitely failed tree of A in P". Generally, the SLD-derivation of A either succeeds, finitely fails or loops. In the annotated completion of the program P, the annotations s and f can be regarded to represent "succeeds" and "finitely fails", respectively. If any attempt to refute A in the program P loops on A then A neither succeed nor finitely fail in the program P.

We show how the annotated completion $AComp(P)$ describes the meaning of logic programs with NF in which inconsistency is included by a simple example.
[Example 1] Let

$$P_A = \{Q \rightarrow R,\ \ Q\} \quad \text{and} \quad P_B = \{\neg Q \rightarrow R,\ \ Q\}$$

be programs that represent each world of agents A and B, respectively. If we take NF as an inference rule of negative information, we have

$$\begin{array}{rcl} AComp(P_A) & = & \{\ (Q{:}s) \rightarrow (R{:}s),\ \ (Q{:}s)\ \} \cup \{\ (Q{:}f) \rightarrow (R{:}f)\ \}, \\ AComp(P_B) & = & \{\ (Q{:}f) \rightarrow (R{:}s),\ \ (Q{:}s)\ \} \cup \{\ (Q{:}s) \rightarrow (R{:}f)\ \}, \\ AComp(P_A) & \models & (R{:}s) \quad \text{and} \quad AComp(P_B) \models (R{:}f). \end{array}$$

Therefore,

$$AComp(P_A) \cup AComp(P_B) \models (R{:}\top).$$

The above formula expresses that the agent A believes "R succeeds" in its world and the agent B believes "R fails finitely" in its world, and there is the contradiction on the predicate R between their worlds. Generally, as $AComp(P_A \cup P_B) \neq AComp(P_A) \cup AComp(P_B)$, the above example does not say that the union of their worlds includes contradiction.

4 Soundness and Completeness of NF

We show the soundness and completeness theorems for NF with respect to the annotated completion. We recapitulate some concepts about ordinary logic programming before giving the theorems.

U_L denotes a Herbrand Universe and B_L denotes a Herbrand Base. We identify as usual a Herbrand interpretation I_H as a subset of B_L. $ground(P)$ denotes the set of all instanciations of clauses in a logic program P. A mapping T_P from a set of Herbrand interpretations to itself is defined as follows : for every Herbrand interpretation I_H,

$$T_P(I_H) = \{A \in B_L \mid L_1 \wedge \cdots \wedge L_m \rightarrow A \in ground(P),\ L_i \in I_H,\ (1 \leq i \leq m)\ \}.$$

Then the finite failure set FF of P and the upward iteration of T_P are defined recursively.

[Definition 3] Let FF_d be a set of ground atoms in B_L which finitely fail in P at depth d [1, 7].

1. $FF_0 = B_L \setminus T_P(B_L)$,
2. $A \in FF_{d+1}$,
   if for every clause, $L_1 \wedge \cdots \wedge L_m \to A$, in $ground(P)$, there is an integer $i (1 \le i \le m)$ such that $L_i \in FF_d$,
3. $FF = \cup_{d \in \omega} FF_d$, where ω is the set of all natural numbers.

The upward iteration of T_P is defined as :

$$\begin{aligned} T_P \uparrow 0 &= \emptyset, \\ T_P \uparrow \alpha &= T_P(T_P \uparrow (\alpha - 1)), \\ T_P \uparrow \lambda &= \sqcup\{T_P \uparrow \eta | \eta < \lambda\}, \end{aligned}$$

where α is a successor ordinal, λ is a limit ordinal, and $\sqcup$ denotes the least upper bound.

A set of Herbrand interpretations can be ordered by the usual set inclusion relation and is a complete lattice. The least fix point of T_P is a set $T_P \uparrow \omega$ and it is equivalent to the least Herbrand model of P.

Now we define a different mapping T_N that maps an interpretation of annotated formulas into itself.

[Definition 4] Let J_H be a Herbrand interpretation of the annotated logic. Then,

$$T_N(J_H)(A) = \sqcup\{ \mu \mid E \to (A:\mu) \text{ is a ground instance of the annotated completion formulas of } A \text{ and } J_H \text{ satisfies the formula} E \},$$

where $\sqcup$ denotes the least upper bound.

We define a special interpretation Δ to be an interpretation which assigns the value l to all members of B_L. Then the upward iterations of T_N is defined :

$$\begin{aligned} &T_N \uparrow 0 = \Delta, \\ &T_N \uparrow \alpha = T_N(T_N \uparrow (\alpha - 1)), \\ &T_N \uparrow \lambda = \sqcup\{T_N \uparrow \eta | \eta < \lambda\}. \end{aligned}$$

[Theorem 1] Let A be a ground atom and P a logic program.

$$A \text{ is a logical consequence of } P \iff P_1^+ \models (A:s).$$

[Proof] Since "A is a logical consequence of P is equivalent to $A \in T_P \uparrow \omega$, this theorem is proved by induction on the least integer d such that $A \in T_P \uparrow d$ and $T_N \uparrow d \models (A:s)$. Let us write $A = A'(t_1, \ldots, t_n)$, where A' is a predicate of arity $n (n \ge 0)$.

Proof of $(\Longrightarrow)$

Basis $d = 1$. In this case, there is a unit clause B in P such that its head can be unified with A and $A \in T_P \uparrow 1$. Then, there is a formula,

$$\forall x_1 \cdots \forall x_n (E_1^+ \vee \cdots \vee E_L^+ \to A'(x_1, \ldots, x_n):s),$$

in P_1^+ such that one of $E_i^+ (1 \le i \le n)$ is $x_1 = t_1 \wedge \cdots \wedge x_n = t_n$. Thus, if P_1^+ is valid in any model of the annotated logic, $A'(t_1, \ldots, t_n):s$ is valid in the model of the annotated

logic.
Induction Hypothesis There is an integer $\alpha \geq 1$ such that ; for any ground atom A, if A is a logical consequence of P, i.e., $A \in T_P \uparrow \alpha$, any interpretation that satisfies P_1^+ is a model of $(A:s)$, i.e., $P_1^+ \models (A:s)$.
Induction Step $d = \alpha + 1$. In this case, there is a clause,

$$B_1 \wedge \cdots \wedge B_m \rightarrow A,$$

in *ground*(P) such that ; for any integer $i(1 \leq i \leq m)$,

$$B_i \in T_P \uparrow \alpha.$$

Thus, by the induction hypothesis, any interpretation that satisfies P_1^+ is a model of

$$(B_1:s) \wedge \cdots \wedge (B_m:s).$$

Then, there is a clause,

$$B_1 \wedge \cdots \wedge B_m \rightarrow A'(t'_1, \ldots, t'_n),$$

in P and

$$P_1^+ \models \exists y_1 \cdots \exists y_k (t_1 = t'_1 \wedge \cdots \wedge t_n = t'_n \wedge (B_1:s) \wedge \cdots \wedge (B_m:s)).$$

Hence,

$$P_1^+ \models A'(t_1, \ldots, t_n):s.$$

Proof of $(\Longleftarrow)$
The proof is by the induction on the least integer d such that a Herbrand interpretation $T_N \uparrow d \models (A:s)$ whenever $T_N \uparrow d$ is a model of P_1^+.
Basis $d = 1$. In this case, we can assume that $P_1^+ \models (A:s)$ and $T_N \uparrow 1$ is a model of P_1^+ and $(A:s)$. Then, there is a formula,

$$\forall x_1 \cdots \forall x_n (E_1^+ \vee \cdots \vee E_L^+ \rightarrow A'(x_1, \ldots, x_n):s),$$

in P_1^+ and there is an integer $i(1 \leq i \leq L)$ such that E_i^+ is $x_1 = t_1 \wedge \cdots \wedge x_n = t_n$. Then, there is a unit clause B in P such that its head can be unified with A and $A \in T_P \uparrow 1$.
Induction Hypothesis There is an integer $\alpha \geq 1$ such that ; for any ground atom A, if $P_1^+ \models (A:s)$, then A is a logical consequence of P, i.e., $A \in T_P \uparrow \alpha$.
Induction Step $d = \alpha + 1$. In this case, there is a clause,

$$B_1 \wedge \cdots \wedge B_m \rightarrow A,$$

in *ground*(P) and a Herbrand model $T_N \uparrow \alpha$ of $(B_1 : s) \wedge \cdots \wedge (B_m : s)$ such that it satisfies P_1^+. Thus, by the induction hypothesis, there is a clause,

$$B_1 \wedge \cdots \wedge B_m \rightarrow A,$$

in *ground*(P) such that ; for any integer $i(1 \leq i \leq m)$,

$$B_i \in T_P \uparrow \alpha.$$

Then, $A \in T_P \uparrow (\alpha + 1)$ and consequently,

$$A \in T_P \uparrow d.$$

Q.E.D.

[Theorem 2] Let A be a ground atom and P a logic program.

$$A \text{ belongs to the finite failure set of } P \quad \Longleftrightarrow \quad P_1^- \models (A:f).$$

[Proof] This theorem is proved by the induction on the least integer d such that $A \in FF_d$ and $T_N \uparrow d \models (A:f)$. Let us write $A = A'(t_1, \ldots, t_n)$ where A' is a predicate of arity $n (n \geq 0)$.

Proof of ($\Longrightarrow$)

Basis $d = 0$. In this case, A cannot be unified with any head of clause in P and $A \in FF_0$. Thus, there is only one clause $A'(t'_1, \ldots, t'_n)$ in P such that its head cannot be unified with A.

Then, there is an annotated completion formula,

$$\forall x_1 \cdots \forall x_n ((x_1 = t'_1 \wedge \cdots \wedge x_n = t'_n) \rightarrow A'(x_1, \ldots, x_n):f).$$

in P_1^- and if P_1^- is valid, $A'(t_1, \ldots, t_n):f$ is satisfied in the model of the annotated logic.

Induction Hypothesis There is an integer $\alpha \geq 1$ such that : for ground atom A, if $A \in FF$, then $P_1^- \models (A:f)$.

Induction Step $d = \alpha + 1$. In this case, for any clause,

$$B_1 \wedge \cdots \wedge B_m \rightarrow A,$$

in $ground(P)$, there is an integer $i (1 \leq i \leq m)$ such that $B_i \in FF_\alpha$. Thus, by the induction hypothesis,

$$P_1^- \models (B_1:f) \vee \cdots \vee (B_m:f).$$

Then, for any clause $B_1 \wedge \cdots \wedge B_m \rightarrow A'(t'_1, \ldots, t'_n)$ in P,

$$P_1^- \models \forall y_1 \cdots \forall y_k (t_1 = t'_1 \wedge \cdots \wedge t_n = t'_n \rightarrow (B_1:f) \vee \cdots \vee (B_m:f)).$$

Consequently,

$$P_1^- \models (A:f).$$

Proof of ($\Longleftarrow$)

The proof is by the induction on the least integer d such that a Herbrand interpretation $T_N \uparrow d$ satisfies $(A:f)$ whenever $T_N \uparrow d$ is a model of P_1^-.

Basis $d = 1$. In this case, we can assume that $P_1^- \models (A:f)$, and $T_N \uparrow 1$ can be a model of P_1^- and $(A:f)$. Then, there is a clause,

$$A'(t'_1, \ldots, t'_n),$$

in P and there is a formula,

$$\forall x_1 \cdots \forall x_n (E^- \rightarrow A'(x_1, \ldots, x_n):f),$$

in P_1^- such that E^- is

$$x_1 = t'_1 \wedge \cdots \wedge x_n = t'_n.$$

Thus, $A = A'(t_1, \ldots, t_n)$ cannot be unified with any head of clauses and $A \in FF_0$.

Induction Hypothesis There is an integer $\alpha \geq 1$ such that : for any ground atom A, if a Herbrand interpretation $T_N \uparrow \alpha$ satisfies P_1^- and $(A:f)$, then $A \in FF_{\alpha-1}$.

Induction Step $d = \alpha + 1$. In this case, for any clause,

$$B_1 \wedge \cdots \wedge B_m \rightarrow A,$$

in $ground(P)$, there is an integer $i(1 \leq i \leq m)$ such that if $T_N \uparrow \alpha$ satisfies P_1^-, then $T_N \uparrow \alpha$ satisfies $(B_i : f)$. By the induction hypothesis, for any clause,

$$B_1 \wedge \cdots \wedge B_n \rightarrow A,$$

in $ground(P)$, there is an integer $i(1 \leq i \leq m)$ such that $B_i \in FF_{\alpha-1}$ and consequently,

$$A \in FF_\alpha.$$

Q.E.D.

5 Remarks

In this paper, we have shown that an annotated logic is appropriate to provide the declarative semantics for NF, although many researchers have already given some logical interpretations to them based on other logics. The main difference between the annotated semantics and the others is the treatment of inconsistency. For example, in the case of a multi-agent system having a contradiction between the belief or knowledge of each agent, we can formalize such a situation of agents by the annotated completion, although we have not mentioned in detail about the treatment of inconsistency.

References

[1] Balbiani,P. : Modal Logic and Negation as Failure. Logic and Computation, **1**, (1991) 331–356

[2] Clark,K.L. : Negation as Failure. (H.Gallair and J.Minker,Eds.), *Logic and Databases* Plenum Press., (1987) 293–322

[3] da Costa,N.C.A., Subrahmanian,V.S. and Vago,C. : The Paraconsistent Logic $P\mathcal{T}$. Zeitschrift für Mathematische Logik und Grundlangen der Mathematik, **37**, (1989) 137–148

[4] Fitting,M. : A Kripke-Kleene Semantics for Logic Programs. J. Logic Programming, **5**, (1985) 295–312

[5] Gabbay,D.M. : Modal Provability Foundations for Negation by Failure. Proc. Int'l Workshop Extension of Logic Programming, LNAI **475**, (1989) 179–222

[6] Kunen,K. : Negation in Logic Programming. J. Logic Programming, **7** (1987) 91–116

[7] Lloyd,J.W. : *Foundations of Logic Programming(2nd edition)*. Springer-Verlag (1987)

[8] Nakamatsu,K. and Suzuki,A. : Automatic Theorem Proving for Modal Predicate Logic. Trans. IECE Japan, **E67** (1984) 203–210

[9] Nakamatsu,K. and Suzuki,A. : Annotated Semantics for Default Reasoning. Proc. PRICAI'94, (1994) 180–186

[10] Nakamatsu,K. and Suzuki,A. : A Non-monotonic ATMS Based on Annotated Logic Programs with Strong Negation. in Agents and Multi-Agent Systems, LNAI **1441** (1998) 79–93

[11] Nakamatsu,K., Abe,J.M. and Suzuki,A. : A Defeasible Deontic Reasoning System Based on Annotated Logic Programming. Proc. 4th Int'l Conf. Computing Anticipatory Systems, AIP Conf. Proc. **573** (2000) 609–620

[12] Nakamatsu,K., Abe,J.M. and Suzuki,A. : Annotated Semantics for Defeasible Deontic Reasoning. Rough Sets and Current Trends in Computing, LNAI **2005** (2001) 470–478

[13] Nakamatsu,K. : On the Relation Between Vector Annotated Logic Programs and Defeasible Theories. Logic and Logical Philosophy, **8** (2002) 181–205

[14] Subrahmanian,V,S. : On the Semantics of Quantitative Logic Program, Proc. 4th IEEE Symp. Logic Programming (1987) 178–182

[15] Thirunarayan,K. and Kifer,M. : A Theory of Nonmonotonic Inheritance Based on Annotated Logic. Artificial Intelligence, **60** (1993) 23–50

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Multi-Agent System for Distribution System Operation

Alexandre Rasi AOKI Ahmed Ali Abdalla ESMIN Germano LAMBERT-TORRES
Electrical Engineering Institute - Federal University of Itajubá
Av. BPS, 1303 – Itajubá / MG – 37500-903 – Brazil

Abstract. There is a permanent demand for new application and simulation software for power systems. The main purpose of this work is the development of computational tool to help operators during restoration task of power substations and distribution systems. This tool was developed using an Object-Oriented approach integrated with the intelligent agent technology. The application of object-oriented modeling in power system has been shown appropriated due to its reuse and abstraction. Furthermore, the Multi-Agent Systems (MAS) are being applied in power system problems solving, and some good results were obtained and the interest in this area has increased in the last two years.

1. Introduction

Due to the current Brazilian power system scenario, an overloaded system in an energy crisis generated by lack of investment and rains, many actions had been taken to overcome this situation. The most prominent action was a plan of control for the energy supply, which lasted for 8 months, and contributed to increase the water storage reservoirs volumes to reach safe levels.

Nevertheless, there is a permanent demand for new application and simulation software, required for different purposes such as research, planning and power system operation. This software becomes larger and increasingly complex, and as a consequence, to create it is more difficult to complete in time and within the budget's constraints. It has become very difficult to create these new applications with traditional software development technology. When finally finished, they are difficult to understand, to maintain, to integrate into old application and to modify for new requirements.

Studies in computer science have shown that the reuse may improve software development productivity and quality. Productivity increases as previously developed assets can be used in current applications, which saves development time. Quality may be increased as frequently reused assets have been tested and corrected in different study cases.

The main purpose of this work is the development of a computer tool to help operators during restoration task of power substations and distribution systems. The restoration of power system normal configuration after a fault, or even a blackout, is performed by intervention of a human operator. Considering the growing complexity in the arrangement of substations and power distribution systems, and the probability of human failure, the time spent in the execution of the restoration actions is larger and has to be optimized.

This tool was developed using an Object-Oriented approach integrated with the intelligent agent technology. The application of object-oriented modeling in power system has been shown appropriated due to its reuse capability and abstraction [1 - 3].

Furthermore, the Multi-Agent Systems (MAS) are being used in power system problems solving, and some good results were obtained and the interest in this area increased in the last two years [4 - 6].

The implementation was done using the Java-based Framework [7, 8], which provides a generic methodology for developing MAS architecture, and a set of classes to support this implementation.

MAS are a way to artificially reproduce real-life system through a model made of autonomous and interacting objects, called agents [9, 10]. The main advantage of multi-agent simulation is to allow the modeling of individual behavior, and the facility to get more real simulation systems. The behavior of the power system components can be simulated by agents that act in the same way.

The development methodology follows five stages: (i) identifying the agents, (ii) identifying the agent conversations, (iii) identifying the conversation rules, (iv) analyzing the conversation model, and (v) MAS implementation. The system developed provides communication, linguistic and coordination support through Java classes. Communication support is provided for both directed communication and subject-based broadcast communication. This feature enables the development of scalable, fault-tolerant, self-configurable and flexible MAS.

2. Multi-Agent Systems

The use of Agent Technology increased over the last decade and several different applications are shown, in order to understand, modeling and develop complex distributed systems, by viewing then as a computational organization consisting of various interacting components [4 – 6, 9, 10]. These applications are founded in different areas, like finance market, Internet, robotics, power systems and educational and simulation problems.

The intelligent agents represent interactive and autonomous independent entities, reproducing real-life phenomena artificially. It is a computer object that has the following properties: autonomy, social ability, reactivity and pro-activeness [9], and besides processing its inputs accordingly to its own intelligence, generating some outputs, Fig. 1.

The agent has the feature of temporal continuity, because it monitors the environment awaiting the occurrences that asks for actions. By analyzing an occurred facts sequence registration, the agent takes the right decision based on its own knowledge. The autonomy feature is given by its decision-making and actions' control to achieve its goals and acquired knowledge based behavior.

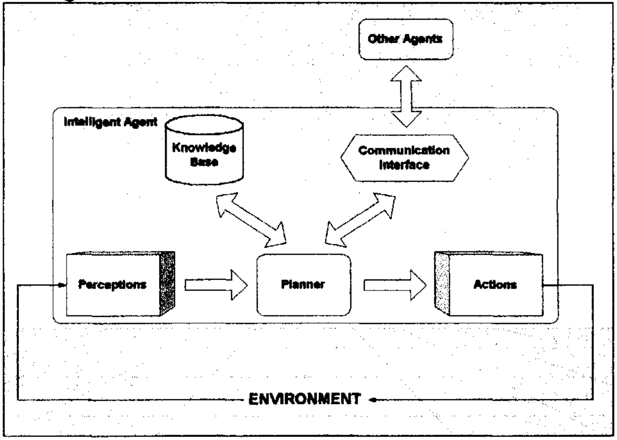


Fig. 1 – Agent Framework

MAS perform collaborative tasks using many intelligent agents in a distributed environment. The Distributed Artificial Intelligence introduces the concept of society to the traditional Artificial Intelligence, using the Object-Oriented approach.

The current studies point to the growing interest on MAS technology for development of complex industrial systems, fragmented in a group of small modules, which is modeled by an Agent or by a small society of Agents.

3. Multi-Agent Model for Power Systems Restoration

The MAS use in power systems research has been increasing with the development of several studies in power systems operation [4], markets [6], diagnosis [11] and protection [12]. The Intelligent Agents' theme for the first time have had a session at The International Conference on Intelligent System Application to Power System, in 2001, and this shows the researchers' interest in this area.

The Multi-Agent Model architecture is shown in Fig. 2. Two main packages and support agents compose this model: a Distribution System Package and a Power Substation Package. There are five support agents working on integration of the model with the real world and among the packages.

3.1 Support Agents

There are five support agents in the model responsible for the integration of model with the real world and among the packages. The Interface Agent is responsible for accessing the SCADA database, so this agent is responsible for the temporal continuity of the MAS in providing timely and consistent information to the other agents. In this agent is described what information is given to each agent, filtering all irrelevant data, minimizing communication and pre-processing tasks.

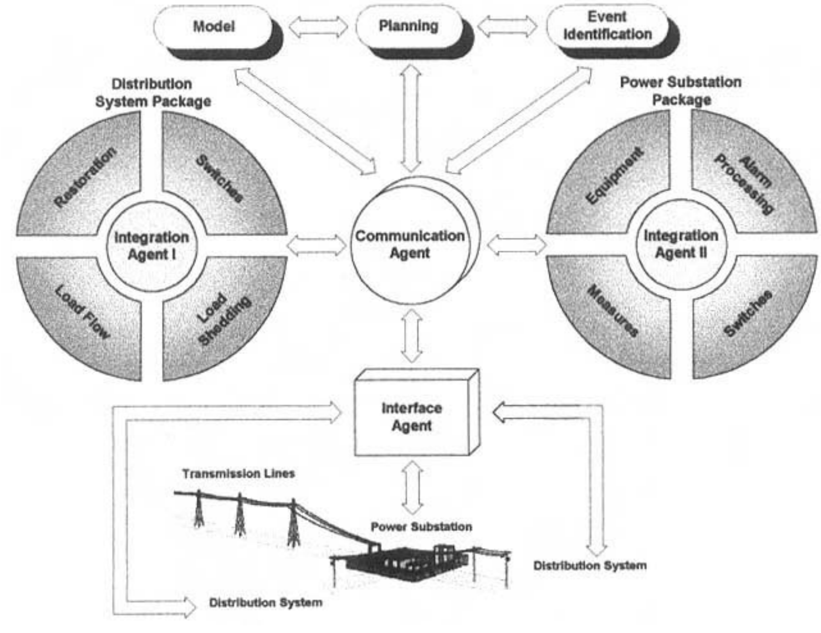

Model
Planning
Event Identification
Distribution System Package
Power Substation Package
Restoration
Switches
Integration Agent I
Load Flow
Load Shedding
Communication Agent
Equipment
Alarm Processing
Integration Agent II
Measures
Switches
Interface Agent
Transmission Lines
Power Substation
Distribution System
Distribution System

Fig. 2 – Multi-Agent Model Architecture

The Communication Agent is responsible for agents' interconnection; it works like communication center: all messages flow through it. This is very important for network processing and for future development of new agents, because the address of a new agent is recorded into the communication agent only, instead of in all the agents.

Event Identification Agent provides expertise to establish the cause-effect relationship between faults and the actions of protective relays and circuit breakers. The implemented strategy detects, initially, a set of fault section candidates, and then, a study about the chronological arrival of relay signals is developed, using the last data set up before the fault. This agent contains a special data where the adjustments of each relay are saved. These data are important to find in what section has started the problem.

Planning Agent is responsible for determining the actions to solve operation tasks. This agent contains a planning algorithm that analysis the event identification agent's information, and all messages from other agents, providing an assigned plan to the user.

Model Agent contains the object-oriented model (Fig. 3) and continuously compares the power system model with the information provided by the Interface Agent. If a difference between the model and the real world is detected the agent immediately inform the communication agent that is responsible for transferring this for the appropriate agent [2, 3].

The object-oriented modeling allows the system's distributed representation, providing flexibility. The distributed representation is interesting for the study of power systems, because of the natural properties of these systems in presenting distributed topology, and also because the model must reproduce the occurred alterations in the real environment. Another advantage presented by this type of modeling is the reuse capability; the structure can be reused for the modeling of another system, being enough to define again the devices and the addition of rules.

3.2 Power Substation Package

The Power Substation Package performs operation tasks in substations, and contains five agents: Alarm Processing Agent, Switching Agent, Measurements Agent, Equipment Agent and Integration Agent I [5].

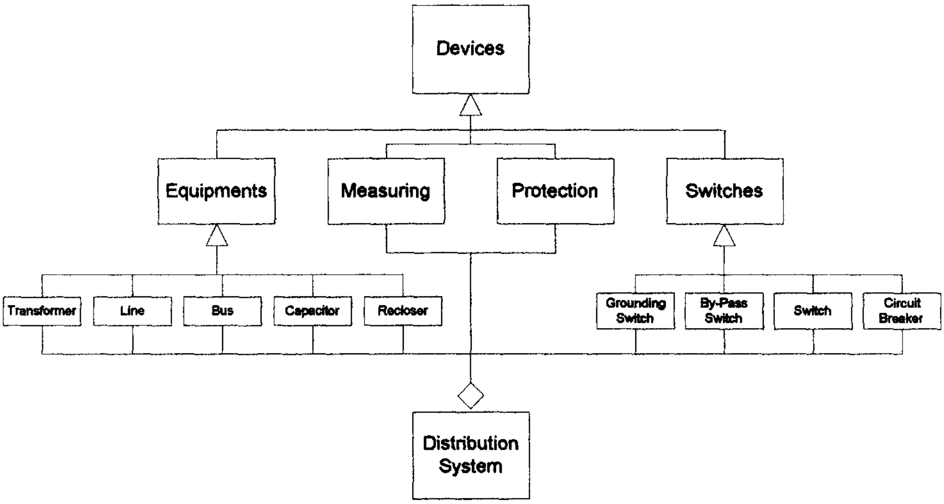


Fig. 3 – Class Diagram of the Distribution System Object-Oriented Model

Alarm Processing Agent provides expertise about the possible occurred problem. When a disturbance occurs in the system, there will be many alarms provided by the SCADA system. Usually, most of these alarms are redundant and happen due to secondary problems caused by the primary problem.

The main idea of this agent is to detect the primary problem, and to send two kinds of alarms to the MAS. The first kind is the alarms of the primary problem, while the second kind are the main alarms for secondary problems. The first kind is very useful for the Event Identification Agent to know the problem and to provide a solution. The second kind (sometimes, more important than the first one) is useful to decide the degree of the contingency.

So, this agent contains two main parts, one for detection of problems and another for evaluation of the disturbance. The first part is composed of production rules, which reads information about the relays and other sensors to define where the disturbance started. The second part makes an evaluation using the data from the files and some rules based on operation conditions.

The Equipment Agent acts as a function of the affected equipment, according to a defined procedure. This program checks up the transformers, buses and capacitors conditions, and in the absence of equipment's inside defects, it signals to liberate the restoration action.

The Measurements Agent monitors the analogical signals of interest, as the voltage values, current, frequency and the angle between voltage and current, in permanent state, with their linked inputs through transducers to CT and PT. In case identification occurs, it stores the post and pre-fault values. This monitoring detects the defect possibilities or oscillations that affect the restoration and, in agreement with the found values, the process can be validated, interrupted or modified, impeding voltage outages and badly energization caused by strategy mistakes and damaged equipment.

The Switching Agent checks up the operated switches, performing continuous monitoring of the switches' state, circuit breakers, grounding switches and by-pass switches. Accordingly with its knowledge base it defines the action to be taken into system switches.

The Integration Agent I is very important for providing integration between packages, and it is responsible for the information of which feeder or buses are operating to the Distribution System Package, or even performs a request for help from an agent of the Distribution System Package, like a load flow analysis of reconfiguration.

3.3 Distribution System Package

The Distribution System Package performs the operation tasks in distribution systems, and contains five agents: Restoration Agent, Switching Agent, Load Flow Agent, Load Shedding Agent and Integration Agent II.

Restoration Agent contains some advice about the best strategy for switching on. Two main ideas provide the ways for problem solving. The first one is to try a restoration by a previously energized feeder. The second idea is to find parallel circuits to provide this restoration. The first idea is possible to apply in radial systems or in temporary unavailable circuits (e.g. temporary faults). In the case where a partial blackout occurs, the system contains a strategy to feed in first place the boundary buses of the blackout system. The idea is to reduce the affected area step-by-step.

Load Flow Agent is a numerical application inside an agent and provides the voltage drop on each feeder branch, the voltage on each bus, and the projected power flow through the distribution system. This information is used by the Planning Agent in finding between the possible solutions, which one has better chances to be executed.

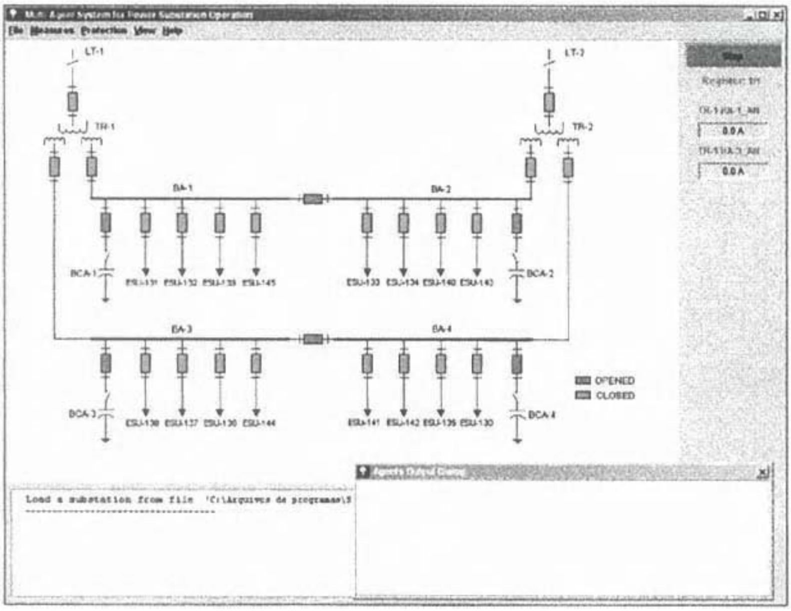
Load Shedding Agent is designed to avoid a frequency or voltage power system collapse. This agent contains a knowledge base about the load shedding strategies of the system under analysis, and comprises a standard numerical program.

The Integration Agent II is responsible for providing integration between packages, for example, sometimes it will be necessary exchange some data for protection analysis of the Distribution System.

4. The Computer Packages

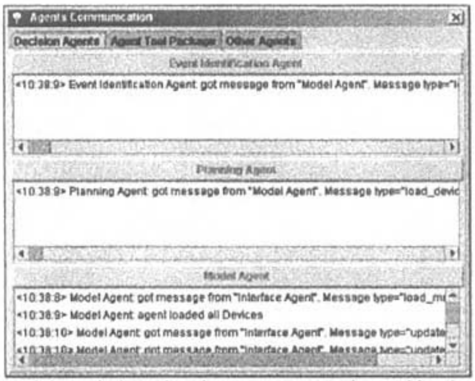
The computer packages are under development, the first one performs the distribution system package tasks, and the second one the power substation package tasks. It was used a Java platform for the power substation problem, as presented in Fig. 4. It has a diagram of the substation, a dialog box, a floating window where plans for operation are presented, and finally a status bar where is presented all system's measures. The user has an option to visualize all protections by activating a menu command, and also, it is possible to visualize all the messages exchanged by the agents, Fig. 5.

For the program of distribution system it was used a Visual Basic platform presented in Fig. 6, and this program is under development yet.


Multi Agent System for Power Substation Operation
File
Measures
Protection
View
Help
LT-1
LT-2
TR-1
TR-2
BA-1
BA-2
BA-3
BA-4
BCA-1
BCA-2
BCA-3
BCA-4
ESU-145
ESU-137
ESU-144
ESU-141
ESU-142
Stop
0.0A
0.0A
OPENED
CLOSED

```
Load a substation from file  'C:\Arquivos de programas\S
--------------------------------
```

Fig. 4 – Power substation program

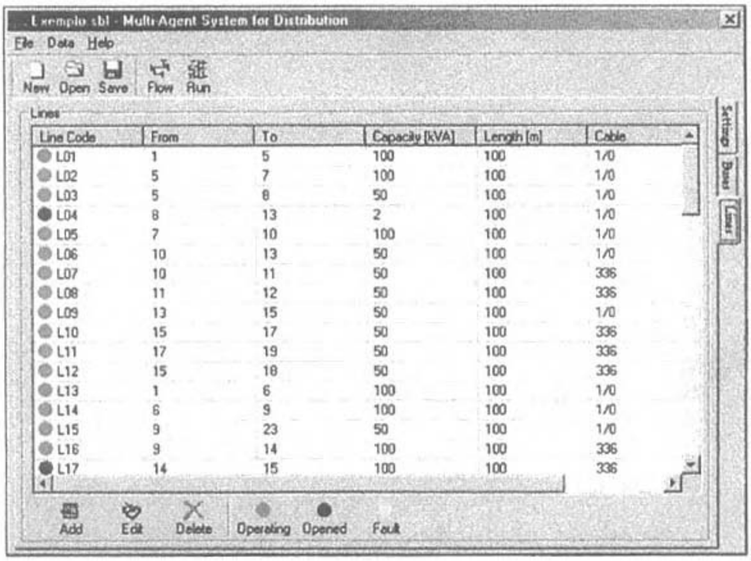

Agent's Communication
Decision Agents
Agent Tool Package
Other Agents
Event Identification Agent
Planning Agent
Model Agent

```
<10:38:9> Event Identification Agent got message from "Model Agent". Message type="l
```

```
<10:38:9> Planning Agent got message from "Model Agent". Message type="load_devic
```

```
<10:38:8> Model Agent got message from "Interface Agent". Message type="load_m
<10:38:9> Model Agent agent loaded all Devices
<10:38:10> Model Agent got message from "Interface Agent". Message type="update
<10:38:10> Model Agent got message from "Interface Agent". Message type="update
```

Fig. 5 – Dialog window showing messages exchanged by the agents


Exemplo sbl - Multi-Agent System for Distribution
File
Data
Help
New
Open
Save
Flow
Run
Lines
Settings
Buses
Lines
Add
Edit
Delete
Operating
Opened
Fault

Line Code	From	To	Capacity [kVA]	Length [m]	Cable
○ L01	1	5	100	100	1/0
○ L02	5	7	100	100	1/0
○ L03	5	8	50	100	1/0
● L04	8	13	2	100	1/0
○ L05	7	10	100	100	1/0
○ L06	10	13	50	100	1/0
○ L07	10	11	50	100	336
○ L08	11	12	50	100	336
○ L09	13	15	50	100	1/0
○ L10	15	17	50	100	336
○ L11	17	19	50	100	336
○ L12	15	18	50	100	336
○ L13	1	6	100	100	1/0
○ L14	6	9	100	100	1/0
○ L15	9	23	50	100	1/0
○ L16	9	14	100	100	336
● L17	14	15	100	100	336

Fig. 6 – Distribution system program

5. Conclusions

The object-oriented model developed has a good degree of abstraction, so it allows the reuse in other topologies. The MAS shows to be adequate for working with the object-oriented model, given to the characteristics of distribution of both the technologies. The main benefits given by the object-oriented model were:

- Distributed representation, allowing the virtual model to reproduce the distributed properties of the system;
- Flexibility, because of the virtual model adaptability to the alterations that may occur representing changes in the real world;
- Reuse capability, using the system developed for modeling a new substation, through the redefinition of the devices and addition or change of rules;
- Open Architecture, allowing addition of components and system expansion, as part of a bigger system.

The MAS architecture is divided into packages of agents that interact to solve the same problem. This packaging is organized as to facilitate the development process, which can be done by different developer groups. In addition, it gives the possibility of including hierarchical analysis of power systems, and flexibility to extend the system through addition of new agents. The main benefits obtained with the multi-agent model were:

- Cooperative behavior, characterizing dynamic exchange of information between entities and making possible segmentation of tasks, besides allowing the reduction of the hierarchic characteristics of the decision process;
- Competitive processing, making possible better use of the hardware resources and reduction of the computational load, improving the execution speed;
- Distributed topology, making possible in the abstraction process, the division of the system in lesser number of parts, limited by functional characteristics;
- Open architecture, allowing the addition of components and the expansion of the system, as part of a bigger system, with creation of new environment levels.

The power substation program was developed and implemented by modulus, in this way it is possible to guarantee flexibility to the expansion and implementation of the multi-agent model. The modular characteristic of the multi-agent systems allows the inclusion of new modules and the expansion of the MAS for power system operation, keeping the characteristics of distribution of the technique.

Beyond restoration tasks, these programs can be used to perform operation tasks like maintenance planning, reconfiguration, etc. This is provided by the general knowledge base developed in each agent.

References

[1] D. Becker, H. Falk, J. Gillerman, *et al*, "Standards-Based Approach Integrates Utility Applications," *IEEE Computer Applications in Power*, Volume 13(4) October 2000, pp. 13-20.

[2] S. Pandit, S.A. Soman and S.A. Khaparde, "Object-Oriented Network Topology Processor," *IEEE Computer Applications in Power*, Volume 14(2) April 2001, pp. 42-46.

[3] S.K. Abidi and A.K. David, "An Object-Oriented Intelligent Approach to System Restoration," in *Proc. ISAP'99*, G. Lambert-Torres and A.P. Alves da Silva, Eds., 1999, pp. 61-65.

[4] M. Amin, "Toward Self-Healing Energy Infrastructure Systems". *IEEE Computer Applications in Power*, Volume 14(1) January 2001, pp. 20-28.

[5] C.R. Lopes Jr., A.R. AOKI, A.A.A. ESMIN, G. Lambert-Torres, "Multi-Agent Model for Power Substation Restoration," in *Proc. IASTED PES 2001*, 2001.

[6] F.-R. Monclair and R. Quatrain, "Simulation of Eletricity Markets: A Multi-Agent Approach," in *Proc. ISAP 2001*, P. Kádár and G. Tarnai, Eds., 2001, pp. 207-212.

[7] Sun MicroSystems, "Implementing Java Computing Solutions White Paper," http://www.sun.cp/nc/whitepapers/.

[8] J.P. Bigus and J. Bigus, "Constructing intelligent agents with Java: a programmer's guide to smarter applications," Wiley Computer Publishing, 1997.

[9] M. Wooldridge and N.R. Jennings, "Intelligent Agents: Theory and Practice," Berlin, Germany: Springer-Verlag, 1994.

[10] J. Feber, "Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence," Addison-Wesley, 1999.

[11] M.A. Sanz-Bobi, J. Villar, *et al*, "DIAMOND: Multi-Agent Environment for Intelligent Diagnosis in Power Systems," in *Proc. ISAP 2001*, P. Kádár and G. Tarnai, Eds., 2001, pp. 61-66.

[12] C.-K. Chang, S.-J. Lee, *et al*, "Application of Multi-Agent System for Overcurrent Protection System of Industrial Power System," in *Proc. ISAP 2001*, P. Kádár and G. Tarnai, Eds., 2001, pp. 73-77.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Ar7bitrariness: Putting Computer Creativity to Work in Aesthetic Domains

Artemis Moroni
Renato Archer Research Center
CenPRA, Brazil
CP 6162 – 13082/120

Jônatas Manzolli
Interdisciplinary Nucleus of Sound Studies
NICS/Unicamp, Brazil

Fernando J. Von Zuben
School of Electrical and Computer Engineering
FEEC/Unicamp, Brazil

Abstract. Ar7bitrariness will be properly formalized here as an interactive iterative optimization process, involving evolutionary computation and other computational intelligence methodologies in the production of user-machine interfaces responsible for alternative mechanisms of aesthetic judgment of computer creativity, particularly on visual and acoustic perceptual domains.

1. Introduction

One of the great powers of computer programming is the ability to define new compound operations in terms of old ones, and to do this over and over again, thus building up a vast repertoire of ever more complex operations. This ability is quite reminiscent of *evolution*, in which more complex molecules evolve out of less complex ones, in an ever-upward spiral of *complexity* and *creativity*. At each stage, the products get more flexible and more intricate, more "intelligent" and yet, more vulnerable to delicate "bugs" or breakdowns [1].

Evolution is now considered useful in simulation to create algorithms and structures of higher levels of complexity. Living things are too improbable and too beautifully designed to have come into existence only by chance. How, then, did they come into existence? The answer, Darwin's answer, is by gradual, step-by-step transformations from simple beginnings, from primordial entities sufficiently simple to have come into existence by chance [2]. Each successive change in the gradual evolutionary process was simple enough, relative to its predecessor, to have arisen by chance. But the whole sequence of cumulative steps constitutes anything but a chance process. When you consider the complexity of the final end-product relative to the original starting point, the cumulative process is directed by non-random survival.

Complexity may be defined as the situation in which, given the properties of the parts and the laws of their interaction, it is not an easy matter to infer the properties of the whole. Also, the complexity of a system may be described not only in terms of the number of interacting parts, but in terms of their differences in structure and function [3]. But there are special problems when examining systems of high complexity: a system may have so many different aspects that a complete description is quite impossible and a prediction of its behavior is unattainable. The analysis of such a system may require the use of approximation models, and there is no efficient procedure to estimate the confidence to be attributed to the final results. There is an enormous complexity in living systems, complexity which does not persist for its own sake, but is maintained as entities at one level, which are compounded into new entities at a next higher level, and so on.

Evolution can be used as a method for creating and exploring complexity that does not require human understanding of the intrinsic processes involved. Interactive evolution normally will depend on a user-machine interface for helping the user with creative explorations, or it may be considered a system attempting to "learn" about human aesthetics from the user. In both situations, it allows the user and computer to work together, in an interactive manner, to produce results that could not be produced alone. Often, creativity in art, literature or science can be understood as a certain *richness of association* with different, perhaps seemingly unrelated disciplines [4]. In the same way, a more creative algorithm would act to go beyond innovation by transferring useful information from other domains.

In what follows, section 2 talks about problem-solvers and computer creativity. Strong and weak problem-solvers are described. In section 3, perceptual selection is associated with interactive genetic algorithms and applied to visual and acoustic domains. Next, the term Ar*T*bitrariness is defined as an interactive iterative optimization process, applied to aesthetic domains. Finally, the conclusions about the consequences of adopting this kind of approach are presented.

2. Problem-solvers and computer creativity

One of the main attributions of the human mind, and certainly a challenge for computational intelligence, is creativity, associated here with the procedures of generating new, unanticipated and useful knowledge, concerning the formal objectives to be achieved in a computational environment. Some problems under investigation are too complex to be properly described and solved using a precise mathematical formalism. The absence of a formal description and the violation of some important restrictions prevent the application of powerful methodologies of solution, denoted here *strong methods*. Strong methods are dedicated problem-solvers that require high-quality and structural knowledge about the problem to be solved, and impose restrictive assumption about the nature of the problem. For example, the best available algorithms for iterative optimization require continuity, convexity, and the availability of second-order information, at each point in the search space. The optimal solution may not be obtained if one of the previous conditions is not valid. So, though being very efficient problem-solvers, strong methods have an undesirable restricted field of application, and the more challenging practical problems of our days cannot be solved by strong methods.

The alternatives are less efficient and generic problem-solvers, denoted here *weak methods*. These methods search for the solution based on a minimum amount of information and restrictions. For example, some algorithms for iterative optimization require neither continuity nor second-order information. So, the field of application of every weak method is much broader than the one associated with its strong counterpart, if available. However, the performance may be completely unsatisfying, because lack of specification associated with the feasible solutions make the space of candidate solutions to be prohibitively large, and the absence of high-quality information prevent the search from being effective.

In rather general terms, there is a compromise between the power of the problem-solver and the required quality of the available information. Nowadays, the most successful tools to deal with this compromise are those inspired in mechanisms and modes of behavior present in intelligent systems [5]. Problem-solvers derived from one (or a hybrid association) of the computational intelligence methodologies guides to something in between strong and weak methods, and that applies creativity in three different manners [6]:

- combinational creativity: definition of an original and improbable combination of already available ideas.

- exploratory creativity: generation of novel ideas by the exploration of structured conceptual spaces.
- transformational creativity: transformation of one or more dimensions of a structured conceptual space, so that new ideas can emerge from the new disposition of information (new topological relations of interest may arise). The level of significance of the dimensions subject to transformation, and the power and adequacy of the transformation, will be responsible for the degree of innovation.

The basic mechanisms involved are information retrieval, association of ideas, evaluation, and selection. In terms of functionality, the synergy of these mechanisms gives rise to analogical thinking, self-criticism and iterative exploration of a well-defined search space. Computer models of creativity include example of all three types, the most successful being those focused in the exploratory creativity. That is not to say that exploratory creativity is easy to reproduce; on the contrary, it typically requires considerable domain expertise and analytical power to define the conceptual space in the first place, and to specify procedures that enable its potential to be explored. But combinational and transformational creativities are even more elusive.

3. Interactive Iterative Reproduction and Perceptual Selection

Programs using evolutionary algorithms can evolve unexpected structures, of a form that the human mind could not have produced by itself. The genetic algorithm is a form of artificial evolution, and is a commonly used method for optimization; a Darwinian 'survival of the fittest' approach is employed to search for optima in large multidimensional spaces. Genetic algorithms allow virtual entities to be created without requiring an understanding of the procedures or parameters used to generate them. The measure of success, or fitness, of each individual can be calculated automatically, or it can instead be provided interactively by a user [7]. Interactive evolution allows procedurally generated results to be explored by simply choosing those having the highest fitness to compose the next generation, so that no mathematical model is necessary to instruct the computer about the evaluation procedure.

Evolution basically consists of endless repetition of *reproduction*. In every generation reproduction takes the genes that are supplied by the previous generation, and hands them on to the next generation with minor random errors (mutations) and recombination of attributes. As the generations go by, the total amount of genetic difference and interchange from the original ancestor can become very large, cumulatively, one small step at a time [2]. But although the mutation and recombination operators incorporate random steps, the cumulative change over the generations is not random. The progeny in any one generation are different from their parent in random directions. But which of those progeny is selected to go forward into the next generation is not random. In every generation, a whole 'litter' of children, or individuals of the next generation, is displayed. All these children are mutant and/or recombinant products of their parents.

3.1 The human eye as the selecting agent

In some aesthetic domains, the selection criterion is not survival, but the ability to appeal to human whim. The human eye has an active role to play in the story: it may be the selecting agent, it surveys the litter of progeny and is asked to determine the fitness of each individual. Then individuals selected for breeding become the parents of the next generation, and a litter of their mutant and/or recombinant children are displayed simultaneously on the screen. This model is strictly a model of artificial selection, not natural selection: the criterion for 'success' is not the direct criterion of survival. So the genes that survive tend automatically to be those genes that confer on bodies the qualities that assist them to survive.

Latham applied in Form Synth the concept of accumulating small changes to help generating computer sculptures made with constructive solid geometric techniques [8]. Sculptors have a number of practical techniques at their disposal to make 3-D forms, for example, welding, chiseling, adding small pieces of clay, wood carving, construction in plastics. The main concept behind Form Synth, however, did not grow out of any existing art style but from another area altogether, the rule-based construction of complex polyhedra from geometric primitives. But on Form Synth the form's evolution entirely depends on the intuitive choices of commands that the user makes, the artist uses the rules successively on geometric primitives and irregular complex forms to produce large "evolutionary tree" drawings of irregular complex forms.

Techniques introduced by Sims [9] contributes towards the solutions to these problems by enabling the "evolution" of procedural models using interactive "perceptual selection". Evolutionary mechanisms of variation and selection were used to "evolve" complex equations used in procedural models for computer graphics and animation. An interactive process between the user and the computer allows the user to guide evolving equations by observing results and providing aesthetic information at each step of the process. The computer automatically generates random mutations of equations and combinations between equations to create new generations of results. This repeated interaction between user and computer allows the user to search hyperspaces of possible equations, without being required to design the equations by hand or even understand them. Sims [7] also successfully applied genetic algorithms to generate autonomous three-dimensional virtual creatures without requiring cumbersome user specifications, design efforts or knowledge of algorithmic details for evolving virtual creatures that can crawl, walk, or even run. The user sacrifices some control when using these methods, especially when the fitness is procedurally defined. However, the potential gain in automating the creation of complexity can often compensate for this lost of control, and a higher level of user influence preserved by the fitness criteria specification.

Figure 1 presents a very simple example of the effect of user-machine interaction in a visual domain. There is no direct intention of including aesthetic affairs here. We implement and evolutionary algorithm such that each individual in the population is a picture composed of 50 lines characterized by color, position of one end point, angle and length. The list of attributes of the 50 lines corresponds to the genetic code of an individual. Twelve pictures are disposed in a uniform grid on the screen, representing all the individuals at the current generation. The task of the user is to attribute a grade to each one of the twelve pictures. After that, the evolutionary algorithm is activated and produces the next generation of twelve pictures. This interactive iterative process goes on until a limit of generations or until a grade level is achieved. The process of grade or fitness attribution should be based on some fitness criterion. In this experiment, the objective is simply to force the lines to be concentrated at the right-bottom corner of the frame. Notice that no explicit instruction is presented to the computer, only better grades are attributed to pictures characterized by a higher concentration of lines at the right-bottom corner. Of course, this rather silly criterion may be replaced by much more complex and meaningful purposes, including the consideration of aesthetic affairs.

In essence, after a number of generations, representing steps of user-machine interaction, the implicit purpose emerges, without explicitly programming the machine to converge to such a proposition. The question is the following: besides guiding lines to the right-bottom corner of a frame, what else can be done? Without taking into account additional practical aspects, the door is now open to include fitness criteria that cannot be describe in mathematical terms or even using a formal language.

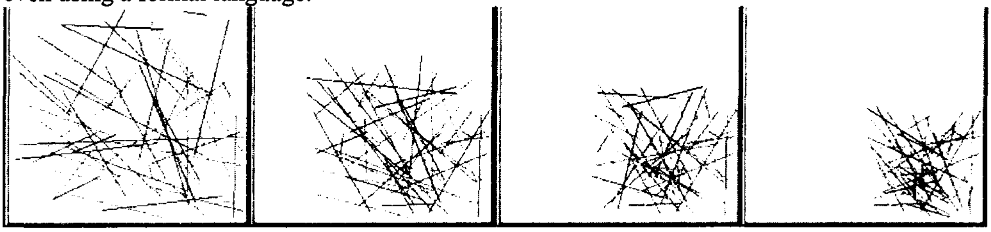


Fig. 1 – Populations of 50 lines were evolved to keep distance from the left and top edges of the frame. From left to right, the pictures corresponds to the individual with the best fitness at generations 0, 4, 8 and 16.

3.2 The human ear as the selecting agent

A subclass of the field of algorithmic composition includes those applications which use the computer as a bridge between an instrument, in which a user "plays" through the application's interface, and as a compositional aid, with which a user experiments in order to generate stimulating and varying musical material. Much of the work that has been done in this field has been based on the idea of determining a set of rules (constraints) which guides the generation of material, rules which are either coded explicitly, or are "learned" by the system from its interaction with the user. Horowitz's development [10] falls into this latter category, given a set of constraining assumptions from which a large number of rhythms can be generated. The system uses an interactive genetic algorithm to learn the user's criteria for distinguishing amongst rhythms. As the system learns (develops an increasingly accurate model of the function which represents the user's choices), the quality of the rhythms it produces improves to suit the user's taste. Interactive genetic algorithms are well suited to solve this problem because they allow a user to simply execute a fitness function (that is, to choose which rhythms he likes), without necessarily understanding the details or parameters of this function. All that a user needs to be able to do is to evaluate the rhythms.

Vox Populi [11] is a hybrid made up of an instrument and a compositional environment. The population is given by groups of four notes, and they are potential candidates to the selection process. The ordering of consonance, i.e. the notion of approximating a sequence of notes to its harmonically compatible note or tonal center, is used. The resultant music moves from very pointillistic sounds to sustained chords, and depends on the duration of the genetic cycle and on the size of the population.

Vox Populi uses the computer and the mouse as real-time music controllers, acting as an interactive computer-based musical instrument. It explores Evolutionary Computation in the context of Algorithmic Composition and provides a graphical interface that allows to change the evolution of the music by using the mouse [12]. These results reflect current concerns at the forefront of interactive composition computer music; nonlinear iterative mappings are associated with interface controls.

The interactive pad supplies a graphical area in which bi-dimensional curves can be drawn. These curves, a blue and a red one, link them with the controls of the interface. The red curve links the melodic and octave range control; and the blue curve links the biological and rhythmic controls. Each curve describes a phase space between the linked variables. They are traversed in the order they were created; their horizontal and vertical components are used for fitness evaluation and to modify the duration of the genetic cycles, interfering directly in the rhythm of the composition. The pad control allows the composer to conduct the music through drawings, suggesting metaphorical "conductor gestures" when conducting an orchestra. By different drawings, the composer can experience the generated music and conduct it, trying different trajectories or sound orbits. The trajectory affects the reproduction cycle and the musical fitness evaluation. Figure 2 presents two different drawings and the generated musical notes resulting from using them as fitness function in Vox Populi.

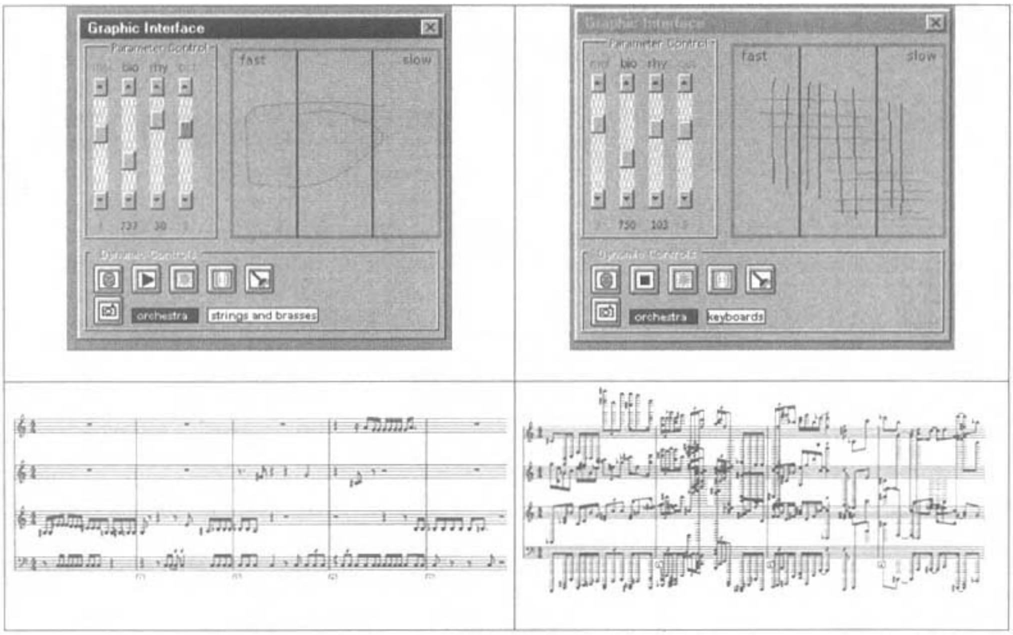


Fig. 2 – In the left, a simple draw and its corresponding musical output generated by Vox Populi system. In the right, a more complex draw and its corresponding musical output.

4. ArTbitrariness

In an interactive genetic algorithm (IGA), human judgment is used to provide fitness, considering a user-machine interface. This cycle typically begins with the presentation of the individuals in the current population for the human mentor to experience them. In visual domains, where each individual typically decodes to an image, all the individuals are usually presented at once, often in reduced size so that the entire population can be contrasted or compared. The mentor can then determine the fitness of each individual with relation to all the others. A well-formalized fitness criterion gives rise to a fitness surface, and a global maximum of this surface in the search space corresponds to the optimal solution. However, to identify the criteria used by the mentor in his evaluation is hard enough. To justify or even explain his reliance on those criteria is still more difficult.

So, the previously described dilemma between strong and weak methods may become even more intricate when the human being is allowed to contribute decisively in one or more steps to be followed by the problem-solver in an exploratory creative domain. As an example, applications of evolutionary computation in artistic domains are often hampered by the lack of a procedure to determine fitness [13]. In such domains, fitness typically reflects an aesthetic judgment determining which individuals in a population are relatively better or worse, based on subjective and often ill-defined artistic or personal criteria.

When the human judgment replaces a formal fitness criterion, we still have a fitness surface, but this surface cannot be expressed in mathematical terms, unless we are able to produce a precise model of the human judgment. Besides, if the human judgment changes with time, than the fitness surface is time-varying. It stands to reason that strong methods have no applicability under these conditions. On the other hand, evolutionary computation techniques remain as powerful methods to implement what we call an *interactive iterative optimization*. In particular, they perform an interactive (fitness criterion defined by human judgment) parallel search for a better solution in a space of candidates (search space), in order to discover promising regions (characterized by the presence of candidates with higher fitness) in the search space and to provide, in average, better solutions at each iterative step (generation), even in the presence of a time-varying fitness surface [14].

When evolutionary computation and other computational intelligence methodologies are involved, every attempt to improve aesthetic judgment will be denoted ArTbitrariness, and is interpreted as an interactive iterative optimization process. ArTbitrariness is suggested in this paper as an effective way to produce art based on an efficient manipulation of information and a proper use of computational creativity to incrementally increase the complexity of the results without neglecting the aesthetic aspects.

However, the potential of ArTbitrariness is not enough to guarantee high performance in every context of application. Roughly speaking, the user should not be requested at the same frequency that the machine is requested. For example, in musical domain not characterized by hybrids like Vox Populi, the temporal evolution of musical events prevents the compressed, parallel presentation of individuals as in computer graphics. Most of the applications of GA to music found in literature presents population as an evolving trajectory of music material such as chords, motives and phrases represented as events. The net result for music, then, is that each member of a population must be presented individually and in real time. This leads to a severe fitness bottleneck, which often limits the population size and the number of generations that realistically can be bred in a musical IGA. These limits are necessary not only to cut down the length of time it takes to run a musical IGA, but also to help reducing the unreliability of human mentors as they attempt to sort through the individuals in a population, listening to only one sample at a time. Nevertheless, mentors often become tired of an overused lick and start punishing individuals in later generations that had been rewarded heavily in earlier generations.

5. Conclusion

Most evolutionary algorithms only explore a pre-given space, seeking the "optimal" location within it, but some also transform their generative mechanism in a more or less fundamental way. For example, evolutionary algorithms in graphics may enable superficial tweaking of the conceptual space resulting in images which, although novel, clearly belong to the same family as those which went before; or it may be so "complexified" that the novel images may bear no resemblance even to their parents, still less to their ancestors. Some should assume that transformation is always creative, or even that artificial intelligent (AI) systems that can transform their rules are superior to those which cannot. Significantly, some AI systems deliberately avoid giving their programs the capacity to change the heart of the code, that is, they prevent fundamental transformations in the conceptual space.

One reason for avoiding rampant transformation in AI-models of creativity is that human may be more interested, at least for a time, in exploring a given space than in transforming it in unpredictable ways. A professional sculptor such as Latham, for instance, may wish to explore the potential and limits of one particular family of 3D-structures before considering others. Another reason is the difficult of automating evaluation. Vox Populi presents the composer with the opportunity to make subjective judgements on all mutations and recombinations while moving to the next iteration. From the composer's point of view it captures the idea of judgement, and from the system point of view it changes the search of musical space as it allows the composer to steer the navigation. The choices interactively made by a composer in response to the evolving music can be stored as a parametric control file and recorded as a musical signal as well. The obtained data can be applied to train neural networks, which in turn may be used as fitness functions, imposing a personal style.

Interactivity, or even real-time performance, is a desired feature in this kind of evolutionary system. Interactivity applied to genetic algorithms means that the user is able to interrupt the evolutionary process at, ideally, any stage and manipulate, ideally, any desired parameter. After such a manipulation, the user must be able to judge the result within a response time that allows interactive working.

Finally, the use of ArTbitrariness is powerful to control the complexity of artistic material in the flow [13]. The relevance of this approach goes beyond the applications per se. Thinking informally, processes that go beyond the innovative are creative. Evolutionary systems that are built on the basis of a solid theory can be coherently embedded into other domains. The richness of associations of the domains are likely to initiate new thinking and ideas, contributing to areas such as knowledge representation and design of visual and acoustic formalisms.

Acknowledgments: Part of this project was possible with the support of FAPESP to the Gesture Interface Laboratory, in which VoxPopuli was developed. Fernando J. Von Zuben is supported by the CNPq grant 300910/96-7. Artemis Moroni is supported by CenPRA.

References

[1] D. R. HOFSTADTER. *Methamagical Themas*, New York: Basic Books, p. 1985.

[2] R. DAWKINS. *The Blind Watchmaker*. London, England: Penguin Books, 1991.

[3] G.A. COWAN, D. PINES AND D. MELTZER (eds.) *Complexity - Metaphors, Models, and Reality*, Santa Fe Institute Studies in the Sciences of Complexity: Perseus Books, Proceedings Volume XIX, 1994.

[4] D. GOLDBERG. "The Race, the Hurdle, and the Sweet Spot", in Bentley, P. (ed.) *Evolutionary Design by Computers*. San Francisco, USA: Morgan Kaufmann, pp. 105-118,1999.

[5] D.B. FOGEL. *Evolutionary Computation - Toward a New Philosophy of Machine Intelligence*, 2nd edition. New York: The IEEE Press, 1999.

[6] M. A. BODEN. "Creativity and Artificial Intelligence", Artificial Intelligence 103, pp. 347 – 356, 1998.

[7] K. SIMS. "Evolving Three-Dimensional Morphology and Behavior", in Bentley, P. (ed.) *Evolutionary Design by Computers* . San Francisco, USA: Morgan Kaufmann, pp. 297 – 321, 1999.

[8] W. LATHAM. Form Synth: The Rule-based Evolution of Complex Forms from Geometric Primitives, in *Computers in Art, Design and Animation* eds. J. Lansdown & R. A. Earnshaw. New York, USA: Springer-Verlag, 1989.

[9] K. SIMS. "Interactive Evolution of Equations for Procedural Models", *The Visual Computer* Vol. 9, No. 9, pp. 466-476, 1993.

[10] D. HOROWITZ. "Generating rhythms with genetic algorithms", *Proceedings of the 1994 International Computer Music Conference*, pp. 142-143, 1994.

[11] A. MORONI, J. MANZOLLI, F. VON ZUBEN & RICARDO GUDWIN. "Vox Populi: Evolutionary Computation for Music Evolution" *in* Bentley, P. (ed.) *Creative Evolutionary Systems*. San Francisco, USA: Morgan Kaufmann, pp. 205 – 221, 2002.

[12] A. MORONI, J. MANZOLLI, F. VON ZUBEN & RICARDO GUDWIN. "Vox Populi: An Interactive Evolutionary System for Algorithmic Music Composition", *Leonardo Music Journal*, Vol. 10, pp. 49-54, 2000.

[13] MORONI, A, VON ZUBEN, F.J. & MANZOLLI, J. ArTbitration: Human-Machine Interaction in Artistic Domains, *Leonardo*, MIT Press, vol. 35, no. 2, 2002.

[14] M.WINEBERG. *Improving the behavior of the genetic algorithm in a dynamic environment*. Ph. D. Thesis. Ontario: Ottawa-Carleton Institute for Computer Science, 2000.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

An Overview of Fuzzy Numbers and Fuzzy Arithmetic

Fernando GOMIDE

State University of Campinas, FEEC-DCA, 13083-970 Campinas, Brazil

Abstract

The aim of this paper is to overview the field of fuzzy numbers and the arithmetic that has been developed to perform operations with fuzzy numbers. Issues concerning overestimation, shape preserving, properties and the expected intuitive characteristics of the operations are particularly emphasized. Recent revisions suggesting how to avoid discrepancies and pitfalls often observed in fuzzy arithmetic are discussed. These issues are of utmost relevance in many engineering, data processing and biological systems modeling and applications because fuzzy numbers are values for fuzzy variables. Moreover, when they are connected with linguistic concepts they provide meaning for values for linguistic variables, a key concept in the theory of fuzzy sets and fuzzy logic.

1. Introduction

In practice, exact values of model parameters are rare in many engineering, data processing, and biological systems modeling and applications. Normally, uncertainties arise due to incomplete or imprecise information reflected in uncertain model parameters, inputs and boundary conditions. This is often the case, for instance, in price bidding in market oriented power system operation and planning, in internet search engines, with the transfer rates in dynamic epidemiological models, and with the amount of carbohydrates, proteins and fat in ingested meals and gastroparese factor in human glucose metabolic models. A fruitful approach to handle parameter uncertainties is the use of fuzzy numbers and arithmetic. Fuzzy numbers capture our intuitive conceptions of approximate numbers and imprecise quantities such as *about five* and *around three and five*, and play a significant role in applications, e.g. prediction, classification, decision-making, optimization and control. In these cases, fuzzy numbers represent uncertain parameters and system inputs, with their support and shape derived from either experimental data or expert knowledge.

Fuzzy quantitites does fit well to many real world cinscunstances, but standard fuzzy arithmetic shows some undesirable effects and may become unsuitable for analysis of uncertain models, especially from the application point of view. For instance, the accumulation of fuzziness, which causes the phenomenon of overestimation, skews the membership functions as a result of fuzzy operations. This effect is similar to that encountered in error accumulation in conventional numerical computations. Fuzzy multiplication and division operations do not guarantee shape preservation. For example, products of triangular fuzzy numbers are not necessarily triangular. Arithmetic operations with fuzzy quantities do, however, satisfy some useful properties as found in conventional operations. They are commutative, associative, but in general they are sub distributive only. Overestimation and shape preserving are particularly problematic because in most cases they mean non intuitive results. Except in pure mathematical and formal contexts, overestimation and lack of shape preservation turn system and model analysis, validation and interpretation difficult.

Overestimation, shape preservation and properties of fuzzy arithmetic are issues constantly debated in the literature and advances have been achieved. One aim of this paper is to emphasize new implementations and recent revisions of fuzzy arithmetic that seem to be relevant for modeling, analysis and applications in engineering, data processing, biology and sytem modeling. We focus on overestimation, shape preservation and properties because they are major components in bringing intuitiveness. We first introduce notation, concepts and definitions to briefly review the standard implementations of fuzzy arithmetic. Next we summarize current efforts that aim to improve on the issues of overestimation, shape preservation and properties of fuzzy arithmetic. The paper concludes addressing issues that deserve further considerations.

2. Fuzzy Numbers and Arithmetic

Fuzzy quantities are intended to model our intuitive notion of approximate intervals and numbers. They are defined in the universe of real numbers R and have membership functions of the form

$$A: R \rightarrow [0,1]$$

A fuzzy number is a fuzzy subset A of R that has the following characteristics [1,2]:

1. $A(x) = 1$ for exactly one x,
2. The support $\{x: A(x)>0\}$ of A is bounded,
3. The α-cuts of A are closed intervals.

It is clear that real numbers are fuzzy numbers. In addition, it can be shown [see e.g. 1, 2] that a fuzzy number is convex, upper semi-continuous, and if A is a fuzzy number with $A(p)=1$, then A is monotone increasing on $[-\infty, p]$ and monotone decreasing on $[p, \infty]$. If the first condition holds for more than one point, then we have a fuzzy interval, figure 1.

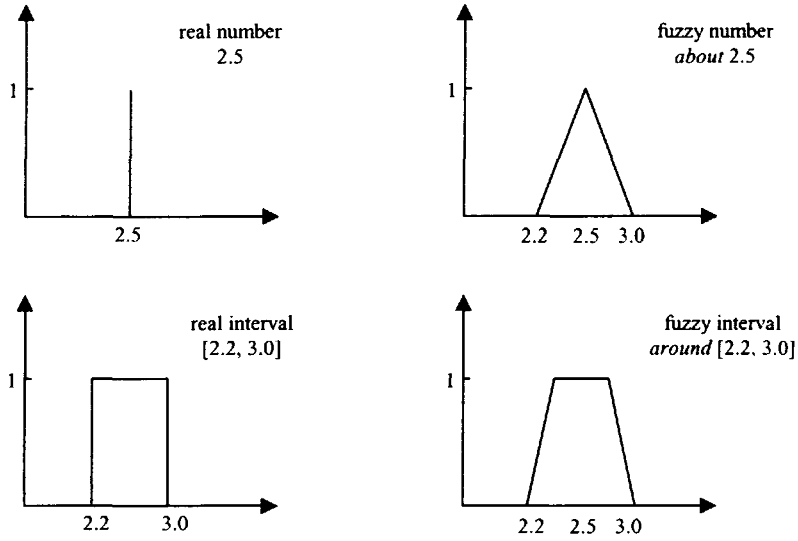


Figure 1: Fuzzy quantities

Basically, there exist two classic methods to perform fuzzy arithmetic operations. The first method is based on interval arithmetic whereas the second employs the extension principle. The extension principle provides a mechanism to extend operations on real numbers to operations with fuzzy numbers.

Let A and B be fuzzy numbers and let $*$ be any of the four basic arithmetic operations. Thus, the fuzzy set $A*B$ is defined via the α-cuts A_α and B_α as $(A*B)_\alpha = A_\alpha * B_\alpha$ for any $\alpha \in (0,1]$. When $* = /$ (division operation), we must require that $0 \notin B_\alpha$ $\forall \alpha \in (0,1]$. Therefore, from the representation theorem [1] we have

$$A*B = \bigcup_{\alpha \in [0,1]} (A*B)_\alpha$$

Thus, the first method to perform fuzzy arithmetic is a generalization of interval arithmetic. The second method uses the extension principle to extend standard operations on real numbers to fuzzy numbers. Therefore the fuzzy set $A*B$ of R is defined by

$$(A*B)(z) = \sup_{z=x*y} \min[A(x), B(y)], \forall z \in R$$

In general, if t is a t-norm and $* : R^2 \to R$ is an operation on the real line, then operations on fuzzy quantities is defined by

$$(A*B)(z) = \sup_{z=x*y} [A(x)\, t\, B(y)], \forall z \in R$$

Figure 2 illustrates the addition $A+B$ of fuzzy quantities A and B for the minimum t-norm t_m and the drastic product t_d t-norm, respectively.

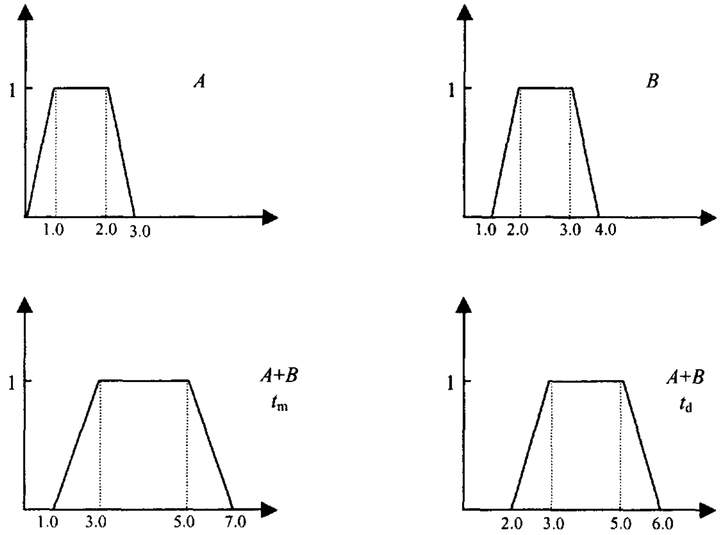


Figure 2: Addition of fuzzy quantities

As surveyed in [3] and more recently in [4], operations with fuzzy numbers do not fulfill some of the properties of the operations with real numbers. A problem with processing fuzzy numbers is related with the validity of group properties and distributivity. This means that $A+(-A)$ is not equal to 0 and $A(1/A)$ is not equal to 1. In addition, if $a, b \in R$, then $(a+b)A$ is not generally equal to $(aA)+(bA)$. Thus $A+A$ does not need to be $2A$, a rather counterintuitive result. Therefore, fuzzy quantities do not form neither an additive nor a multiplicative group if strict equality is demanded, and 0 and 1 are considered as zero and unit element. This complicates some theoretical considerations and practical procedures. See [3, 4] for further details on these issues.

Direct implementation of fuzzy arithmetic via interval arithmetic generally is computationally complex. Implementation of the extension principle is equivalent to solve a nonlinear programming problem. To overcome this difficulty, often we limit to simple membership functions such as triangular or trapezoidal as depicted in figure 1. In these cases, approaches to compute fuzzy operations become simpler because they can be done via the parameters that define the fuzzy numbers. Unfortunately, the shape of triangular numbers is not preserved under certain operations. More specifically, the shape of the triangular fuzzy numbers is not closed under multiplication (see figure 3) and division because the result of these operations is a polynomial membership function and triangular approximations can be quite poor and produce incorrect result [5]. This is particularly crucial in engineering and biological systems modeling and applications. Next we shall discuss the main efforts that have been addressed to overcome some of the problems inherent to classic fuzzy arithmetic.

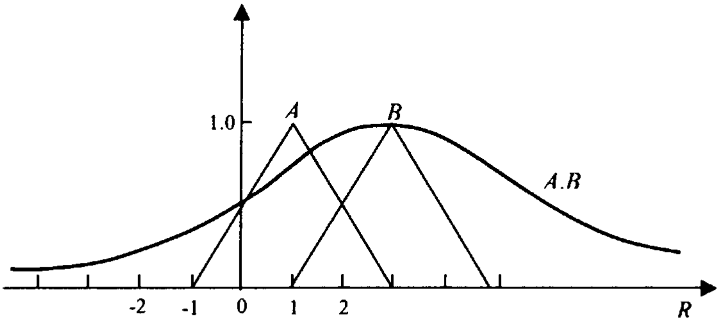


Figure 3: Multiplication of two triangular fuzzy numbers

3. Revisions and New Approaches for Fuzzy Arithmetic

In what follows, we review new methods to perform fuzzy operations and produce more intuitively meaningful results. In general, the methods assume, either explicitly or implicitly, a form of constrain in the operation procedure. Properties such as associativity, may be lost and further computational difficulties may arise. Context dependent heuristics seems to provide a powerful and pragmatic approach to turn fuzzy arithmetic consistent with human intuition and practicalities.

3.1. Requisite Constraints

When arithmetic operations are performed with real numbers we implicitly assume they are independent of the objects they represent. This principle is also assumed in the usual interval and fuzzy arithmetic. However, usually they are not independent as they are tacitly tightened to a theoretical or application context. For example, if we let $A=(1,2,4)$ be a triangular fuzzy number whit modal value 2 and left and right base points 1 and 4, respectively. Then its α-cut is $A_\alpha=[1+\alpha, 4-2\alpha]$ and $(A-A)_\alpha=[-3+3\alpha, 3-3\alpha]$. This result does not fit our intuition and may lack practical meaning. The point is that fuzzy set operations (subtraction in the example) neglect the fact that the operands are equal. Requisite constraints is a key concept recently been introduced by Klir [6] to address this issue. Clearly, as shown in the example above, it is essential to include the equality constraint, when applicable, into the general definition of basic arithmetic operations. Otherwise, we may get results that are less precise and counterintuitive. In general, as suggested in [6], arithmetic operations constrained by a relation $\Re$ becomes, using the generalization of interval arithmetic and the extension principle,

$$(A*B)^R{}_\alpha = \{x*y/\ (x,y) \in A_\alpha \times B_\alpha \cap \Re_\alpha\}$$

$$(A*B)^R(z) = \sup_{z=x*y} \min[A(x), B(y), \Re(x,y)]$$

respectively. Fuzzy arithmetic with requisite constraints is a fruitful area, with mathematical and computational challenges yet to be resolved before it becomes fully operational.

3.2. Discrete Fuzzy Arithmetic

Motivated by the general practice to represent analog functions by its sampled version for computational purposes, Hanss [7,8] suggests a discrete approach to represent and to operate on fuzzy numbers. Fuzzy numbers can be made discrete discretizing either the universe or the membership space, the unit interval for instance. Discretization of the universe is unsuitable because the shape of the membership function varies with the discretization level. This can be avoided if the membership space is subdivided into a number of m, equally spaced segments $\Delta\mu=1/m$. Thus, a fuzzy number can be considered either as approximated by a discrete fuzzy number, or as being decomposed into intervals $[a_i, b_i]$, $a_i \leq b_i$, $i=0,1,\ldots,m$ given by the α-cuts at the α-levels μ_j with $\mu_j=\mu_{j-1}+\Delta\mu$, $j=1,..,m$, $\mu_o=0$, $\mu_m=1$. Then, given n independent fuzzy numbers A_i, $i=1,\ldots,n$, each of them can be decomposed into a set of $m+1$ intervals X_i^j , $j=0,1,..,m$ of the form $[a_i^j, b_i^j]$, $a_i^j \leq b_i^j$, $i=0,1,\ldots,n$, $j=1,..,m$. Assuming that the problem is to find the arithmetical expression of the form $Q=F(A_1,..,A_n)$, its evaluation is then performed by evaluating the expression separately at each of the 2^n positions of fields using conventional arithmetic. A field is a transformation of the intervals to form 2^{i-1} pairs. See [8] for further details. The implementation suggested by Hanss can be viewed as multiple evaluations of different sets of crisp values nested according to their level of membership. As pointed out in [8], if F is monotone function, then the result of the operations do represent a proper solution at every level of membership. Otherwise, the result might show a slight difference from the proper result. The difference decreases as m increases.

3.3. Specialized Approaches

A natural way to model approximately known values such as numbers that are around a given real number, is fuzzy arithmetic. From the common sense point of view, around hundred plus one is still around hundred ($\approx 100 + 1 \approx 100$), although in fuzzy arithmetic this is not the case. A modification of fuzzy arithmetic that does retain the common sense view has recently been proposed by Kreinovich and Pedrycz [9]. Intuitively, their method assumes that, given two fuzzy numbers A and B, we should compute the membership function of $C=A+B$, find the interval of possible values of C (e.g. as the values such that $C(x) \geq p_o$ for some value p_o, pick the simplest value c on this interval, and then return "around c" as the result of adding A and B. The key here is what does *simplest* mean. The formalism introduced in [9] suggests the shortest length of a formula $F(y)$ in a language L which defines the particular number x, (i.e. which is true for $y=x$ and false otherwise) as a measure of complexity $D(x)$ of a real number x.

The current brain and biological information processing knowledge has indicated that counting and operations with cardinal quantities is an ability encountered in many animals. This ability seems to be essential in their struggle to survive and evolve. Being a complex distributed information-processing system, the brain dynamics suggests mechanism to perform arithmetic operations in distributed organizations. Here, fuzzy numbers play an essential role once they are part of the effort to process approximate quantities as a means to gain competitiveness. In this vein, Massad and Rocha [10] have suggested the definition of K-fuzzy numbers, a number whose precision depends of the number itself. The reader is referred to the reference [10] for details.

4. Analysis and Discussion

Most, if not all of the current efforts to bring fuzzy arithmetic closer to meaningful and intuitively understandable results, as required by models of real world systems, still lack formal and computational analysis. The requisite constraint approach [6] does provide an appropriate framework to verify the properties and characteristics of the arithmetic operations. However, it is not clear yet what properties are these and which algorithms are relevant for efficient computational implementations. The discrete fuzzy arithmetic approach [8] has indications to perform computationally well and to be semantically consistent with the expected results. However, arithmetic operations properties still are to be characterized. The common sense viewpoint of fuzzy arithmetic has been shown to lack associativity and to be difficult to compute [9]. Shape preserving is not, in general, guaranteed by any method. Recent research on fuzzy operations with t-norms has provided hints on what classes of t-norms are tuned with shape preservation for some operations. The discrete fuzzy arithmetic approach claims to provide a solution to the overestimation problem. In fact, the examples discussed in [7, 8] do suggest that this is the case, but generally speaking, formalization of overestimation still is an open issue.

5. Conclusions

The importance of fuzzy numbers is unquestionable as witnessed in many application areas. Mathematically speaking, fuzzy arithmetic is well developed. However there are questionable results due to a deficient generalization from arithmetic on reals to fuzzy numbers in particular and fuzzy quantities in general. This generalization ignores constraints induced by the application context, a key issue suggested in practice, but still operationally negleted. Consensus on a proper generalization still is to be found and is a considerable challenge. Heuristics seems to be a key concept to be explored. In particular, approximate reasoning indicates a path in which mathematical properties can be jointly considered with human perception to provide intuitive and meaningful operations. An alternative is to use fuzzy rasoning as an approach to link fuzzy systems modeling and fuzzy arithmetic operations as suggested in [11].

Acknowledgements

The author acknowledges CNPq, the Brazilian National Research Council, for its support.

References

[1] G. Klir and B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall, Upper Saddle River, 1995.

[2] H. Nguyen and E. Walker, A First Course in Fuzzy Logic. CRC Press, Boca Raton, 1997.

[3] D.Dubois and H. Prade, Fuzzy Numbers: An Overview. In: J. Bezdek (ed.) Analysis of Fuzzy Information, CRC Press, Boca Raton, 1988, vol. 2, pp. 3-39.

[4] M. Mares, Weak Arithmetics of Fuzzy Numbers. *Fuzzy sets and systems*, **91** (1997) 143-153.

[5] R. Giachetti and R. Young, A Parametric Representation of Fuzzy Numbers and their Arithmetic Operators. *Fuzzy sets and systems* **91**(1997) 185-202.

[6] G. Klir, Fuzzy Arithmetic with Requisite Constraints. *Fuzzy sets and systems*, **91** (1997) 165-175.

[7] M. Hanss, On Implementation of Fuzzy Arithmetical Operations for Engineering Problems. *Proceedings of NAFIPS 1999* (1999) 462-466, New York.

[8] M. Hanss, A Nearly Strict Fuzzy Arithmetic for Solving Problems with Uncertainties. *Proceedings of NAFIPS 2000* (2000) 439-443, Atlanta.

[9] V. Kreinovich and W. Pedrycz, How to Make Sure that "≈100"+1 is ≈100 in Fuzzy Arithmetic: Solution and its (Inevitable) Drawbacks. *Proceedings of the FUZZ-IEEE* (2001)1653-1658, Melbourne.

[10] E. Massad and A. Rocha, Implementing Mathematical Knowledge in a Distributed Intelligent Processing System. *Proceedings of the conference IPMU'2002*, Annecy, July 2002 (to be published).

[11] D. Filev and R. Yager, Operations on Fuzzy Numbers via Fuzzy Reasoning. *Fuzzy sets and systems*, **91** (1997) 137-142.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

The Brain and Arithmetic Calculation

F. T. Rocha and A. F. Rocha

School of Medicine, University of São Paulo, LIM01/HCF-MUSP
Av. Dr. Arnaldo 455, São Paulo, 01246-903, SP, Brazil
eina@enscer.com.br

Abstract

This paper describe the results of an experimental study about the cerebral activity associated with and the time required for arithmetic operation solving in adults and children. The main findings show that males are faster than females in solving arithmetic calculations, independent of age and degree of education. This difference in performance is supported by different neuronal recruitment in man and woman.

1. Introduction

Counting is a process depending on a distributed number representation in the [1, 2], and whose efficacy is mainly dependent on the optimization of the control of the eyes and hands to focus the objects to be counted [1, 3].

Counting is also proposed to be the underlying process for arithmetic calculation since it have been demonstrated that the time (response time) required to produce the result of an arithmetic calculation is dependent on the size of its numbers [2, 5 - 7].

It has also been demonstrated that different strategies may be used by the same person to solve the very same calculation [7]. For instance, addition may be solved by finger marking of all elements of the sets being added (full manipulation strategy); or by finger marking only the elements of the smallest of these sets (minimum manipulation strategy); or by no opened finger manipulation (mental manipulation strategy) [6, 7].

It has also been reported that man and woman differ on their arithmetic capabilities [8, 9], perhaps boys being faster than girls in retrieving mathematical facts, what could explain why male outperforms females in timed arithmetic tests [9].

Many different areas of the brain have been demonstrated to be involved in counting and arithmetic calculations [1, 2]. Neurons in the inferior parietal cortex, angular gyrus and prefrontal areas are among those most frequently enrolled in arithmetic manipulations[1, 10, 11]. Some authors have also presented some experimental evidences claiming sex differences on neuronal enrollment for arithmetic processing.

The purpose of the present paper is to investigate the chronometry and neuronal backing of the arithmetic operations in two groups of volunteers: children attending the second degree of the elementary school and adults enrolled in a technology master program.

2. Methodology

Experimental group **A** was composed by 8 girls and 8 boys attending the second year of the elementary school, mean age around 8 years. Experimental group **B** was composed by 10 male and 10 female enrolled in a technology master course, mean age around 30 years.

Each volunteer solved 30 arithmetic calculations similar to that in Fig. 1, automatically presented in the video of a Pentium III®, while having their electroencephalogram (EEG) recorded. The computer automatically measured the time (**RT**) elapsed since the visually presentation of the calculation to be performed up to the moment the volunteer selected a number at the left side as the solution of the arithmetic operation. The computer also tracked possible errors. Each volunteer was asked to solve 30 additions, 30 subtractions, 30 divisions and 30 multiplications.

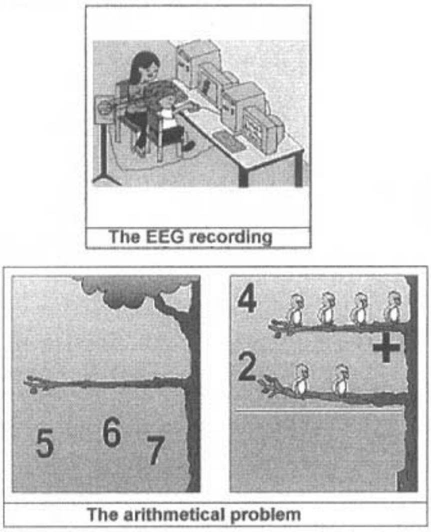


Figure 1: The experiment

The recorded EEG was analyzed according to the technique described by Foz et. al. [12]. Numerical data were analyzed according to each group, sex and mathematical operations. Non-parametric statistics was used to evaluated mean sex differences. The following index was used to quantify sex differences [7]:

$$I = RT_f - RT_m / SD_g$$

where RT_f, RT_m stand for the mean response time for females and males in the group g and SD_g stands for the standard deviation calculated for the each experimental group. Factor Analysis was used to disclose principal cerebral activity components associated with each arithmetic calculation and sex.

3. Results

The statistical analysis of the data showed a clear difference between sexes for the group of adults (B in Fig. 2). Males were faster than females in solving all arithmetic operations. These differences were greater for addition and division, and were minimum for product.

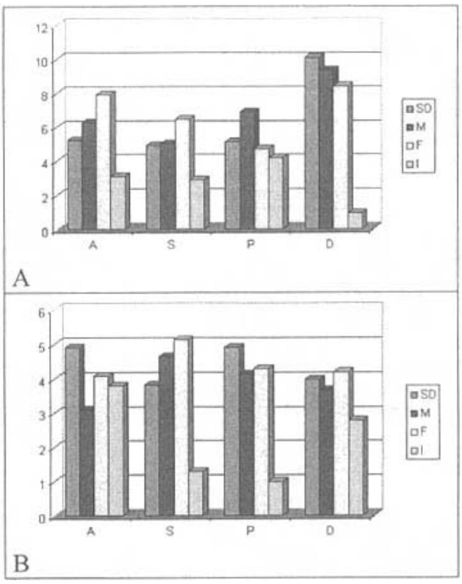
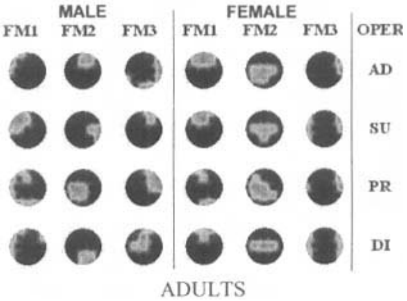


Figure 2: Statistical results for RT mean values

A: adition, S: subtraction, P: product and D: division. SD: group standard deviation, M: mean male RT, F: mean female RT, I: index of sex differences.

Sex differences were also observed for the group of children (A in Fig. 2), but they were not statistically significant in the case of division. But it must be said that division is the arithmetic operation being learned during the second year of the elementary school. The sex differences in group A were greater for the product and almost the same for addition and subtraction.

Response time for children were greater than that for adults for all arithmetical operations. No important differences were observed for the different arithmetic operations in both experimental groups.



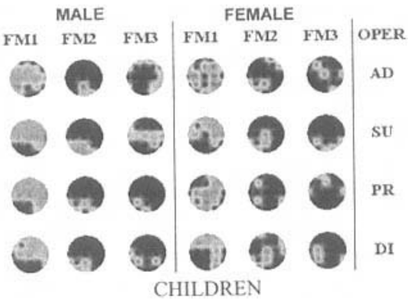


Figure 3: Cerebral activity - Principal Components

AD: Addition, Su: Subtraction; PR: Product; DI: Division; FM1, FM2, FM3: the three principal components identified by factor analysis.

Factor analysis disclosed three principal components of cerebral activity associated with arithmetic calculations in both experimental groups (Fig. 1); which accounted for around 30%, 30% and 20% of data variance, respectively. The brain areas associated to each factor varied according to the calculation being performed, sex and experimental group. Despite this, some general patterns of brain activation may be assumed to exist in each experimental group.

In the case of male adults, FM1 tended to involve left hemisphere areas in all types of calculation, while it tended to group bilateral frontal regions in the case of female adults. FM3 tended to involve right hemisphere areas in both groups, but also the left hemisphere in the females. FM2 tended to be much more similar considering the different arithmetic calculations for females than for males. Women tended to use neurons on central and parietal areas more frequently than men, which in turn tended to recruit more the right hemisphere. Sex factor differences were less pronounced for product than for the other types of calculations.

In the case of children, FM1 involved a huge number of areas in both hemispheres, such that sex differences may appear more blur than for the other two factors. FM2 tended to group central parietal and left occipital areas in both boys and girls, but it also recruited right neurons in females. FM3 seems to be the factor that shows the greatest sex differences in the case of children. It also exhibits a greater variety when the different types of operations are considered.

4. Conclusion

The present results seem to clearly demonstrated a sex difference on arithmetic calculation when the time required to obtain the desired result is considered and the cerebral activity is analyzed.

The difference in performance seems to be established as early as the beginning of the elementary school and persists despite academic training. But, performance improvement from infancy to adulthood is accompanied by a clear modification of the neural circuits supporting all types of arithmetic calculations.

Rocha and Rocha [13] showed that the solution of arithmetic calculations does not seem to depend on some verbal memorization, since RT depended on the size of numbers involved in the calculations and that the use of different solving strategies, that may change according the academic training.

The present results seem to confirm those assumptions since factor analysis disclosed correlated activity mostly among areas that are classically associated to non-verbal processing. Most of the areas evidenced to participate on arithmetic calculations are distributed over the right hemisphere and the occipital lobe, which area regions assumed to be mostly involved with visual processing [10, 14, 15]. Other frontal and parietal areas appearing here to be associated to calculus solving, have being proposed to be involved with the hand and eyes movement control [16, 17].

The present results seem also to show that people may use different strategies to solve the same arithmetic problem. The variability of the brain patterns disclosed by the factor analysis when the type of calculation, sex and age are considered, may be the reflex of using different neural circuits to achieve the same result. For instance, maybe adult males relied more upon visual operations supporting block counting as could be evidenced by a substantial use of the right hemisphere showed by FM2 and FM3; while females used more frontal and parietal neurons as showed by FM1 and FM2, to simulate a mental sequential (finger) counting.

The performance improvement from infancy to adulthood may be associated to the dramatic change on the style of brain activation showed by factor analysis when children and adults are compared. Children seems to enroll many neurons widely distributed over the brain to solve all types of calculations, as showed by FM1, whereas no such pattern is exhibited by any of the factors in the case of adults. Also, it seems to exist a greater variability of FM2 and FM3 for all arithmetic calculations when children are compared to adults.

The present results and those published by Rocha and Rocha [13] seem to support the conclusion that the different strategies used to solve arithmetic problems, are supported by a counting process that did not depend on language, but relied a distributed processing system specialized for quantification and calculation as proposed by Massad and Rocha [3,4].

References

[1] B. Butterworth, The mathematical brain, Macmillan, London, 1999.

[2] S. Dehaene, The number sense, Penguin Books, London, 1997.

[3] E. Massad. and A. F. Rocha, Implementing arithemetical knowledge in a distributed intelligent processing system. To appear in Proceedings of IPMU, 2002a.

[4] E. Massad and A. F. Rocha, Evolving arithmetical knowledge in a distributed intelligent processing system. To appear in this Proceedings, 2002b.

[5] M.H. Ashcraft, Cognitive arithmetic: A review of data and theory. In Numerical Cognition, Dehaene (Ed); Elsevier, Amsterdan, 1991, pp. 75-106

[6] M. Fayol, Lénfant et le nombre: du comptage à la résolution de problémes, Delachaux & Niestlé, Paris, 1990.

[7] R.S. Siegler, Emerging minds Oxford Univesity Press, London, 1996.

[8] D.F. Halpern, Sex differences in cognitive abilities. Laurence Erbaun, Hillsdale, 1992, pp. 216-221.

[9] J.M. Royer, L. Tornsky, H. Marchan and S.J. Jackson, Math_Fact retrieval Hypothesis, Contemporary Educational Psychology, 1999, pp. 24:286-300.

[10] F.R. Fink, J.C. Marshall, J. Gurd, P.H. Weiss, O. Zafiris, N.J. Shah and K. Zilles, Deriving numerosity and shape from identical visual displays NeuroImage 13, 2001, pp.46-55.

[11] S. Gobel, V. Walsh and M.F.S. Rushworth, The mental number line and the human angular gyrus. NeuroImage, 23, 2001, pp. 1-10.

[12] F.B.F Foz, L.P. Luchini, S. Palmierie, A.F.Rocha, E.C Rodela, A.G. Rondó, M.B. Cardoso, P.B. Ramazznin, and C.C Leite, Language plasticity revealed by electroencephalogram mapping. Pediatric Neurology, 2002, pp.106-115.

[13] A.Rocha and F.T. Rocha, Mental processes of arithmetic calculation. To appear in Proceedings of IPMU 2002.

[14] F. Chochon, L. Cohen, P.F. van de Moortele and S. Dehanene, Differential contributions of the left and right inferior parietal lobules to number processing. J. Cog. Neuroscience. 11, 1999, pp.617-630.

[15] K. Sathian, T. J. Simon, S. Peterson, G.A Patel, J.M. Hoffman and S.T. Grafton, Neural evidence linking visual object enumeration and attention. J. Cog. Neuroscie. 11, 1999, pp. 36-51.

[16] R.A. Anderson, Multimodal integration for the representation of space in the posterior parietal cortex In The Hippocampal and Parietal foundations of spatial cognition., Buergess, N, Jeffery, K. J and O'Keefe, J (eds), Oxford University Press, 1999, pp. 90-101.

[17] C.R. Olson, S.N. Gettner and L. Temblay. Representation of allocentric space in the monkey frontal lobe. In The Hippocampal and Parietal foundations of spatial cognition., Buergess, N, Jeffery, K. J and O'Keefe, J (eds) Oxford University Press, 1999, pp.357-380.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Evolving Arithmetical Knowledge in a Distributed Intelligent Processing System

A. F. Rocha and E. Massad

School of Medicine, University of São Paulo, LIM01/HCF-MUSP
Av. Dr. Arnaldo 455, São Paulo, 01246-903, SP, Brazil
eina@enscer.com.br and edmassad@usp.br

Abstract

A new class of fuzzy numbers, called K Fuzzy Numbers, (KFN) is proposed to implement arithmetical knowledge in a Distributed Intelligent Processing System (DIPS). The proposed system intends to simulate brain functioning and experimental data about arithmetic capability in man and animals. KFN main property is the dependence of the size of the base α_1, α_2 of its membership function μ_{di} (σ) to the value of the number d_i encoding σ. Three populations of (Controllers; Accumulators and Quantifiers) agents are assumed to handle KFN in DIPS. Also, it is proposed that Accumulator evolution changing from monotonic to periodical evaluation strategies, may be the way to use KFN to create Crisp Base Numbers (CBN), like our decimal numeric system.

1. Introduction

A Distributed Intelligent Processing System (**DIPS**) is composed by [1 - 3]:

a) a finite set of agents, each of them instrumented to solve a given class of problems, and
b) a finite set of resources to support communication transactions among these agents,

such that the solution of a complex task becomes a job for a group of agents recruited according to the suitability of their tools to handle the given problem.

DIPS knowledge ought to be spatially distributed rather than being the privilege of one or few agents. Resulting redundancy then turns this knowledge resistant to corruption, and it is mainly implemented by a set of agents having similar but not identical tools to handle de same piece of information.

The solution of any (new) task different from those already solved by the system, is to be pursuit by attempting either novel agent recruitment or new agent specialization. Therefore, DIPS' intelligence is both determined by the plasticity of communication transactions and evolution of agent specialization.

The brain is the most complex DIPS known, and special attention has recently being paid to understand its mathematical capabilities [4, 5]. Basic to the issue is the process of quantification of the numerosity or the cardinality of a set. Human being call this process by the name of *counting* and claim it to be its privilege. However, neurosciences have being demonstrated that many animals are also able to quantify the cardinality of those sets (of food or predators) important to their survival, and arithmetic calculation has been proposed to evolve from cardinality quantification [6, 7].

The purpose of this paper is to discuss how this counting and arithmetic capabilities may be implemented in a DIPS and how they may evolve from simple systems as those used by animals to our complex mathematical knowledge.

The paper is organized as follows. Section 2 and 3 describes the DIPS model introduced by Massad and Rocha [8, 9] to account for cardinality quantification as proposed by neurosciences. Section 4 proposed how changes on the strategies used by some DIPS' agents may result in counting evolution from animals to modern humans. Section 5 makes some comments on this evolution.

2. A model of a Quantifier DIPS

Let the following set of agents be part of a DIPS labeled **K**:

a) a finite set **S** of sensory agents having tools to sensor the universe *U*; that is each $s \in S$ is equipped with tools of the type $m = f(v)$, where m is a measure about the variable v; and
b) a finite set **R** of recognition agents specialized in identifying objects o_i belonging to *U*, as sets Ψ_i of relations between the measures m_i sensored by **S** about these o_i. In this condition, the image $I(o_i)$ used by **R** to identify o_i is defined as the minimum set of the relations that are uniquely associated to o_i and to no other o_j in *U*; that is $I(o_i) \subset \Psi_i$.

Many objects o_i may be simultaneously identified by **R** as a consequence of the redundancy inherent to any DIPS. Let β_i be the maximum number or block of o_i that may be simultaneously identified by **R**. This is a very primitive way of quantifying o_i in *U*, because it is dependent and limited by agent redundancy in **K**. Also, this process is circumscribed to the subspace F of *U*, called the sensory field of **S**, simultaneously monitored by its sensor agents. This quantification process is named *subitizing* by neurosciences, and it is considered a special kind of counting [4, 5].

The improvement of the identification of the cardinality of any set O_k composed by the same type of objects o_j in *U*, requires the participation of other types of **K** agents. This cardinality identification process will be called here *cardinality quantification*, or **CQ** for short.

First of all, **CQ** requires a sequential recognition process to be implemented in order to overcome the limitation imposed by block identification (Fig. 1 and 2). This may be obtained with:

a) a finite set **C** of control agents in charge of moving **S** over *U*, in order to cover the subspace V containing o_i, that is

$$V \subseteq \{ F_j \}_{t=1 \text{ to } u} \qquad (1)$$

where F_t is the subspace of *U* sensored by S at the step **t**, whose size is F and **u** is the number of steps required to cover V;

b) a finite set **A** of agents in charge of accumulating the quantities of o_i identified at each step of the sequential covering of V. Each agent $a \in A$ performs computations of the type

$$\sigma_{t+1} = \sigma_t - \pi * \beta(1 - \omega/t), \; \omega > 1 \qquad (2)$$

and t as above, σ the actual accumulated value and β is the quantity of o_t identified at the step $t + 1$;

c) a finite set **Q** of agents that are able to recognize the quantities accumulated by **A**, by classifying their readings of $\boldsymbol{\sigma}$. In this, each $\mathbf{q}_i \in \mathbf{Q}$ performs computations of the type

$$\upsilon_i = \mathbf{d}_i \text{ if } \alpha < \sigma < \beta \text{, otherwise } \upsilon_i = \eta \quad (3)$$

where υ_i is the output of $\mathbf{q}_i$; $\alpha,\beta > 0$; $\mathbf{d}_i$ is a label in a dictionary **D**, η is the empty label in **D.**

As knowledge must be spatially distributed in **K**, the representation of an increased (and perhaps infinite) number of quantities in **K** must rely on spatially ordered distribution of the finite set **Q**. The optimization of this representational process requires also the maximal number Ψ (perhaps equal to ∞) of identified quantities to be greater than the number ρ of agents in **Q**.

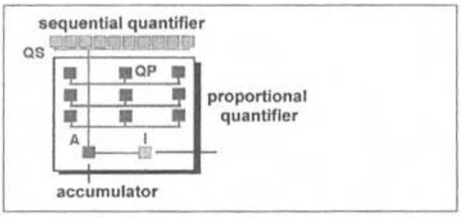


Figure 1: The proposed DIPS named K

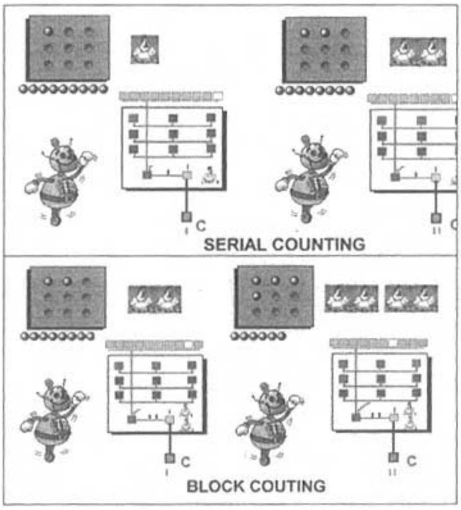


Figure 2: Quantifying (or counting) the cardinality of sets of objects

The spatial ordered distribution of **Q** may be support by:

a) proportional quantification: if the quantity to be represented depends on the number of recruited **q** agents, or

b) sequential quantification: if the quantity to be represented depends on the site occupied by its specialized agent.

Now, the quantification of the cardinality of any set $\mathbf{O}_k$ composed by the same type of objects $\mathbf{o}_j$ distributed over $V \subseteq U$ is obtained by sequentially controlling the covering of V by F (Fig. 2). This may be achieved by two different strategies: unitary or block object identification. In the first case, the sensors of **S** are focused over each object $\mathbf{o}_j$ and the result of its identification is unitarily accumulated by **A.** Whenever the spatial distribution of $\mathbf{o}_i$ in V allows up to β objects $\mathbf{o}_j$ to be simultaneously identified, then the quantification may proceed by block accumulation in **A** of the total of the identified elements.

CQ optimization is mostly dependent on the improvement of the control of the pathway of S to cover V. This optimization requires:

a) the same object or block to be focused just once, and
b) if time is constrained, the covering pathway to be short as possible.

So, **CQ** optimization may involve complex computations, similar to those required to solve popular AI problems, such as the travelling sales man.

3. K Fuzzy Number

The **CQ** capacity to handle sets of large cardinalities depends on the implementation and development of adequate Sequential Quantifier Systems (**SQS**) allowing the maximal quantified cardinality to be greater than the number of quantifiers, by encoding cardinalities by means of agent relations.

Let the distribution of data provided by **A** to **Q,** be governed by filtering the information transmitted by the channels used to support communication transactions among their agents.

Let this function to be denoted by f and to be dependent on the geographical location g of $\mathbf{q}_g \in \mathbf{Q}$, such that

$$\mathbf{q}_g \text{ at } \mathbf{g} \text{ attends to } \mathbf{A} \text{ if } \alpha_g < \sigma < \beta_g \tag{4}$$

In other words:

$$\upsilon_g = \mathbf{d}_g \text{ if } \sigma - \alpha_g < \sigma < \sigma + \alpha_g, \tag{5}$$

otherwise $\upsilon_g = \eta$, (6)

$$\mathbf{d}_g \in \mathbf{D} = [\, \mathbf{d}_1, \ldots, \mathbf{d}_n, \eta \,] \tag{7}$$

and

$$\alpha_g, \beta_g = f(\mathbf{g}) \tag{8}$$

Now, let the following rewriting rules $\oplus$, $\otimes$ be used to order **D:**

$$\mathbf{d}_{i+1} = \mathbf{d}_i \oplus \mathbf{d}_1, \mathbf{d}_1 = \mathbf{d}_1 \oplus \eta \tag{9}$$

and

$$\mathbf{d}_{i-1} = \mathbf{d}_i \otimes \mathbf{d}_1, \mathbf{d}_1 = \mathbf{d}_1 \otimes \eta \tag{10}$$

then, a special case of **SBS** is defined if

$$\upsilon_g = \mathbf{d}_g \text{ if } \mathbf{d}_g \otimes \gamma_g < \sigma < \mathbf{d}_g \oplus \gamma_g, \tag{11}$$

otherwise

$$\upsilon_g = \eta, \tag{12}$$

$$d_g \in \dot{D} = [\, d_1, \ldots, d_n, \eta \,] \tag{13}$$

$$\gamma_g = K(\, d_g \,)\,, \gamma_g \in D \tag{14}$$

Now, if the possibility μ_{di} (σ) of σ to be encoded by d_i may is defined as

$$\mu_{di}\,(\sigma) \rightarrow 1 \text{ if } \sigma \rightarrow d_i \tag{15}$$

$$\mu_{di}\,(\sigma) \rightarrow 0 \text{ if } \sigma \rightarrow d_i \otimes \gamma_i \text{ or } d_i \oplus \gamma_i \tag{16}$$

and K is a sigmoid function then **D** may be partitioned into a subset

$$D_c = [\, d_1, \ldots, d_j \,] \tag{17}$$

of crisp numbers; another subset

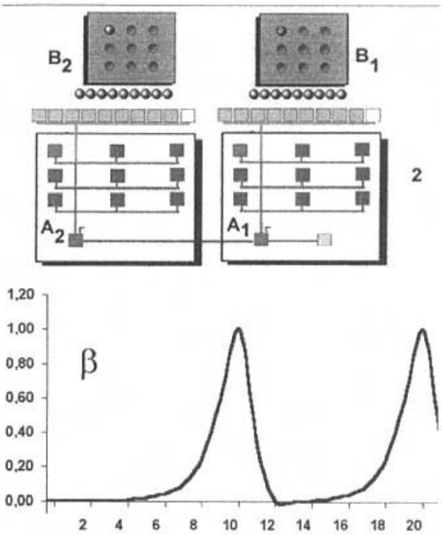
$$D_f = [\, d_k, \ldots, d_l] \tag{18}$$

of fuzzy numbers a third subset

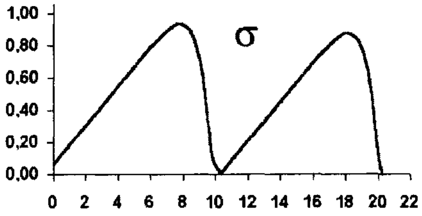
$$D_c' = [\, d_l, \ldots, d_n \,] \tag{19}$$

which is crisp but deals with blocks of orders of magnitudes.

This type of SQS is called here **K Fuzzy Numbers (KFN)**.



3a



3b

Figure 3a and 3b: Evolving KFN counting

4. Crisp Base Numbers

Now let be assumed an evolution of the **K DIPS** introduced in section 2, characterized by increasing the complexity of the population of Accumulators and Quantifiers (Fig. 3), obtained by:

a) changing from the linear accumulation defined by eq. 2 to a non-linear function such as

$$\beta_{t+1} = Z_1 * \sigma_t * \beta_t^L \qquad (20)$$

$$\sigma_{t+1} = \sigma_t - Z_1 * Z_2 * \sigma_t * \beta_t^L + Z_3 \qquad (21)$$

as depicted by fig. 3;

b) creating a hierarchy of subsets of accumulators ($A_1, A_2, \ldots$) to monitor the discontinuities on the lower level accumulators;

c) creating a hierarchy of sequential quantifiers (B_1, $B_2, \ldots$) associated to that of accumulators (see fig. 3), and

d) associating the monotonic increasing accumulation to the crisp subset

$$\mathbf{D_c = [\, d_1, \ldots, d_j \,]}$$

in **17**.

This type of DIPS is able to handle a set of Crisp Base Numbers (**CBN**) whose size of block is defined by the crisp set in **17**. Our decimal numbers are the special case of CBN with

$$\mathbf{D_c = [\, 1, \ldots, 9, 0 \,]} \qquad (22)$$

5. Conclusion

The model proposed here incorporates many of the basic properties of the neural systems involved in quantifying the cardinality of sets [4, 5] and defines a special class of fuzzy numbers, called here **K Fuzzy Numbers**, for which the base of their membership function is a function of the number magnitude. The properties of **KFN**s nicely map upon the experimental data obtained for both animals and man in cardinality set estimation.

But more important, it is the fact that **KFN** circuits may evolve to other **DIPS** having the capability of handling crisp numbers, or Crisp Base Numbers (**CBN**). This evolution may be attained by

a) changing the linear accumulation function used by some of the **DIPS'** agents into a non-linear estimation;

b) evolving different subsets of accumulators and quantifiers, and
c) adequate pairing of these subsets of accumulators and quantifiers.

The biological evolution experienced from other animals to man may be understood as consequent to gene modification allowing:

a) first the change from linear to periodical accumulation functions used by some cells located at multi-sensorial areas, such as the parietal lobe, and
b) second by increasing the number of available accumulators and quantifiers.

The genetic evolution may allow the two types of accumulators to exist in the brain, maybe the most primitive normally predominating if learning of cultural numerical transmitted system does not occur [9]. This could explain the use of KFN by children and many types of indian cultures.

The discovery of **CBN** by humans was certainly favored by the genetic evolution creating non linear accumulators. However, the transmission of this discovery must have been guarantee by memetic inheritance [9]. Such inheritance may be proposed to be supported by learning by imitation or teaching. Both mechanism may recruit the biochemical transactions required to activate the reading of the genes for non-linear accumulation and if necessary to block the expression of the genes for linear estimation [10].

It may be concluded, therefore, that the model presented here to implement arithmetic knowledge in a **DIPS,** not only incorporates many of the basic properties of the neural systems involved in quantifying the cardinality of sets, but also may explain the evolution from **KFN** to **CBN**, that have occurred in the human specie.

References

[1] W. Pedrycz, P.C.F. Lam and A.F. Rocha, Distributed Fuzzy system Modeling. IEE Trans. Sys. Man Cyber. 25, 1995, pp. 769-780.
[2] A.F. Rocha, The brain as a symbol processing machine, Progress in Neurobiology 53, 1997, pp. 121-198.
[3] A.F. Rocha, A. Pereira Jr. and F.A.B. Coutinho, NMDA Channel and Consciousness: from signal Coincidence Detection to Quantum Computing, Progress in Neurobiology 64, 2001, pp. 555-573.
[4] B. Butterworth, The mathematical brain, Macmillan, London, 1999.
[5] S. Dehaene, The number sense, Penguin Books, London, 1997.
[6] M. Fayol, Lénfant et le nombre: du comptage à la résolution de problémes, Delachaux & Niestlé, Paris, 1990.
[7] R.S. Siegler, Emerging minds Oxford Univesity Press, London, 1996.
[8] E. Massad. and A. F. Rocha, Implementing arithmetical knowledge in a distributed intelligent processing system. To appear in Proceedings of IPMU, 2002a.
[9] E. Massad and A. F. Rocha, Meme-gene Coevolution and Cognitive Mathematics. To appear in this Proceedings, 2002b.
[10] Pereira Jr, A. (2002) Neuronal Plasticity: How Memes Control Genes. To appear in this volume.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Meme-Gene Coevolution and Cognitive Mathematics

Eduardo Massad and Armando F. Rocha

School of Medicine, University of São Paulo, LIM01/HCF-MUSP
Av. Dr. Arnaldo 455, São Paulo, 01246-903, SP, Brazil
edmassad@usp.br, eina@enscer.com.br

Abstract

After presenting the basic aspect of the new science of memetics and the basic mechanism of meme-gene coevolution we discuss the role of memes in the evolution of brain size and complexity. We also address the issue of cognitive mathematics, that is, how the human brain deals with number and present some hypothesis on the evolution of our mathematical abilities. We end the paper with a brief discussion of the future perspectives of memetics and cognitive mathematics, arguing that this is a very rich field of investigation, still in its infancy.

1. Introduction

Uniquely among animals, humans are capable of imitation and so can copy from one another ideas, habits, skills behaviors, inventions, song and stories. These are all *memes*, a term which first appear in Richard Dawkins' book *The Selfish Gene* [1]. In that book, Dawkins dealt with the problem of biological (or Darwinian) evolution as *differential survival of replicating entities*. By replicating entities Dawkins meant, obviously, genes. Then, in the final part of his book, Dawkins asked the question '*are there any other replicator on our planet?*', to which he answered '*yes*'. He was referring himself to cultural transmission and fancied another replicator – a unit of imitation [2]. Dawkins first though of '*mimeme*', which had a suitable Greek root (Dawkins' words) but he wanted a monosyllable word which would sound like 'gene' and hence the abbreviation of mimeme – meme. A revolutionary new concept (actually, a truly Kuhnian paradigm shift) was born. Like genes, memes are replicators, competing to get into as many brains as possible.

One of the most important memes created by humans is the concept of numbers. As mentioned by Dehaene [3] numbers are cultural inventions only comparable in importance to agriculture or to the wheel. Counting, however, is a process observable in a great number of non-humans species. Millions of years earlier the first written numeral was engraved in bones or painted in cave walls by our Neolithic ancestors, animals belonging to several species were already registering numbers and entering them into simple mental computations [3]. We, modern humans differ from the other species by being able to deal with numbers in a highly sophisticated manner, rather than the simple block-counting process characteristic of lower species which is typically limited to 3 or 4 units.

The purpose of this paper is to introduce the logic and basic assumptions of memetics, presenting its definition and presenting memetics as a potential answer to the problem of brain evolution and the human capacity of dealing with complex mathematical processes.

2. The subject matter of memetics and its developments

The *Oxford English Dictionary* contains the following definition:

> **meme** *An element of a culture that may be considered to be passed on by non-genetic means, esp. imitation.*

Memes can be thought as information patterns, held in an individual's memory and capable of being copied to another individual's memory. The new science of memetics is a theoretical end empirical science that studies the replication, spread and evolution of memes.

As the individual who transmitted the meme continues to carry it, the process of meme transmission can be interpreted as a replication, which makes the meme as a truly replicator in the same sense as a gene. Likewise the evolution of traits by the natural selection of those genes which confer differential reproductivity, the cultural evolution also occurs by selection of memes with differential reproductivity, that is, those memes with the highest copying rates will take over those with lower copying rates.

Dawkins listed three characteristics for any successful replicators: *copying-fidelity* – the more faithful the copy, the more will remain of the initial patterns after several round of copying; *fecundity* – the faster the rate of copying, the more the replicator will spread; and *longevity* – the longer any instance of the replicating patterns survives, the more copies can be made of it. Just think of the example provided by Blackmore [2], the song 'Happy Birthday to You' and you have a tremendously successful replicator, already copied (with high-fidelity) thousand of millions of times (high fecundity) all over the world for several decades (longevity).

In these characteristics, memes are similar to genes and the new science of memetics imitates (a metamemetics phenomenon?) to a certain extent, that of genetics.

3. Memetics and cultural evolution/gene-culture coevolution

Memetic and genetic evolution can interact in rich and complex ways, a phenomenon described as 'meme-gene-coevolution [2]. Cultural evolution is a branch of theoretical population genetics and applies population genetics models to investigate the evolution and dynamics of cultural traits equivalent to memes [4]. Gene-culture coevolution employs the same methods to explore the coevolution of genes and cultural traits. Meme evolution, in turn, may occur either exclusively at the cultural level, or through meme-gene interaction, both mechanisms having important consequences. In the following sub-sections I intend to present examples of those consequences of meme dynamics.

One important concept to the following arguments is that of meme-gene coevolution. According to this theory, social learning and memetic drive are the cause behind the evolution of increased brain size [4]. According to Blackmore [2], this process has three stages. In the first, individuals with a genetic predisposition for imitating would succeed over those that only learn directly from the environment. This argument assumes that the object or the behaviour imitated increases the performer's fitness. In the second stage, in a population of imitators, those with a genetic predisposition to imitate from the best imitators (selective imitators) would be selected. Thirdly, assortative mating between selective imitators would produce the most successful offspring who would inherit the genetic predisposition to imitate selectively and the imitation gene frequency would increase in this population. As selective imitation requires a greater than average cognitive capacity there is, therefore, a selective pressure for an increase in the average brain size. In addition, a minimally sophisticated language system is required for meme spreading, which also contributes to the increasing in brain size. Hence, the meme-gene coevolution acting sinergically to the evolution of our sophisticated language and enormous brain.

4. Memetics and the evolution of the human brain size

Since, unlike genes, memes do not come package with instructions for their replication, our brain do it for them, strategically, guided by a fitness landscape that reflects both internal drives and a world view that is continually updated through meme assimilation [5].

The increase in brain size began with the macroevolutionary events that culminated with the first species of the Homo genus, about two and a half million years ago. By about 100,000 years ago *H.sapiens* had brains as large as ours and the other sapiens sub-species, the Neanderthals had brains with a volumetric capacity larger than ours. They controlled fire, had cultures and probably had some form of language as well.

The increase in brain size, however, have a price [2]. Oversized brains are expensive to run and ours consumes 20% of the body energy for a size correspondent to only 2% in weight. In addition, brains are expensive to build. The amount of protein and fat necessary for the development of human brain forced the first members of the Homo gender to increase their meat consumption, which implied in better hunting strategies, which in turn feed back in increased brain size. Finally, big brains are dangerous to produce. The increasing in brain size, along with the bipedal gait of Homo specimens implied in severe birth risk. The big brained human babies have enormous difficulty to pass through the narrowed birth canal. This implies, in addition to the higher maternal and foetal mortality, that the human baby born prematurely, as compared with other primates. On one hand this have the beneficial consequence that our brain has greater neuronal plasticity, which increases its learning capacity. On the other hand, the complicated twisting manoeuvres the human foetus had to do in order to pass through the birth canal implies that the human female rarely is able to deliver without assistance. This also contributes to socialisation and additional selection for brain increasing.

Our brains have changed in many ways other than just size. The modern human prefrontal cortex, oversized when compared with other hominids, is fed by neurons coming from practically all other parts of the brain. Its role in the complex cognitive abilities of modern humans is still to be decifrated but we already known that when damaged by accident or surgical removal (a common practices some decades ago) the victim is severely limited in its calculation performance (in addition to many other personality changes). We will return to the prefrontal cortex later on this text.

So why did the human brain increased so much? Several theories have been proposed but in this paper we will adopt the meme theory proposed by Blackmore [2]. According to this author the turning point in our evolutionary history was when we begun to imitate each other. This implies that the selection pressures which produced the increase in our brain size were initiated and driven by memes, according to the following rationale. Suppose that your neighbour has learned (or invented) some really useful trick or tool. Then, by imitating him you can avoid the slow, costly and potentially dangerous process of trying out new solutions by yourself. It is obvious the competitive edge imitation can have. Imitation, however, requires 3 skills [6]:

a) making decision about what to imitate;
b) complex transformation from one point of view to another; and
c) production of matching bodily actions.

Imitation is, therefore, a tremendously demanding task. If imitation provides a competitive edge, it is, therefore, natural to expect selection pressures to increase the organ responsible for this phenotypic characteristic – the very brain. This selection pressures happen in four steps [6]:

1) selection for imitation;
2) selection for imitating the imitators (genes for imitating the best imitators will increase in the gene-pool due to the selective pressures);
3) selection for mating with the imitators; and
4) sexual selection for imitation. This last step is based on Blackmore`s assumption that females can increase the number of their descendants in future generation by choosing mates who will give them attractive sons who will have many offspring.

In summary, big brains is a far from expected natural event in our evolutionary path, due to the heavy constrains they impose. Therefore, the genecomplex responsible for increase in brain size must provide a selective advantage to its possessors. The memetic theory of brain size assumes that imitation was the driving force behind the selective pressures – genes have been force into creating big brains capable of spreading memes.

5. Cognitive mathematics: how the human brain deals with numbers

We are born with a capacity to enumerate objects which is strictly limited to very few items and we share this genetically determined characteristic with more `primitive` (that is, less brained) animals. Several empirical evidences have already demonstrated that the counting capacity is innate in rats, pigeons and monkeys [3]. There is even an observable and remarkable competence of human babies in simple arithmetic. This genetically determined arithmetical competence was studied by several authors since the beginning of the last century. Some interesting experiments by early psychologists were summarised by Dehaene [3] in which human subjects are asked to enumerate objects. The results are that enumerating a collection of items is fast when there are one, two or three items, but starts slowing drastically beyond four. In addition, errors begin to accumulate at the same point. To circumvent this genetical upper bound in our counting capacity we invented a clever strategy: counting in blocks, a process called in the specialised literature as `subtizing`. It takes about five- or six-tenth of a second to identify a set of three dots , or about the time it takes to read a word aloud or to identify a familiar face and this time slowly increases from 1 to 3 [3]. Therefore, subtization requires a series of visual operation all the more complex the greater the number to be recognised. But we are endowed with a highly sophisticated thinking machine whose sheer size and tremendously complex wiring allows us to outstanding cognitive performance without parallel in the phylogenetic scale.

Even Neanderthals, whose brain were equal or slightly bigger than ours, but whose neuronal composition and configuration were probably strikingly different from ours, could not imagine to reach the complexity of our cerebral products, like modern mathematics. How and why this cognitive abilities evolved? It is tempting to answer this question by the obvious assumption that the larger the brain the more its owner capacity to adapt and survive in aggressive and/or rapidly changing environments. However, as mentioned above, large brains are expensive if overdimensioned and the role of memes in the evolution of brain size was already explained in the previous section. One could then argued that, in addition to the size of the brain, it is its configuration which is responsible for the differential mathematical competence of modern humans. But is there a cerebral region responsible for mathematical thinking? The first experiments, carried out in the early eighties, demonstrated a higher cerebral activity during numerical performance, in particular the inferior parietal cortex as well as multiple regions of the prefrontal cortex [7]. Recent experiments with functional magnetic resonance, however, demonstrated that several other cerebral areas are activated during mental calculations [3]. It is now accepted that the inferior parietal region is important for the transformation of numerical symbols into quantities, and the representation of relative number magnitudes. The extended prefrontal cortex in turn is responsible for sequential ordering of successive operations, control over their execution, error correction, inhibition of verbal responses, and, above all, working memory. It is impossible to dissociate memory from calculation!

6. How did our mathematical competence evolved?

Let us imagine the African environment of about 150,000 years ago. The first wave of *Homo erectus* emigration out of Africa had already occurred. Groups of Neanderthals wandered throughout Europe and Asia, endowed with a cerebral capacity of the same order of magnitude of modern humans but, as mentioned above, with a neuronal configuration certainly different from us. At that time, the first modern humans were organising themselves into tribes and, the cognitive threshold of meme spreading had already being surpassed. These humans certainly had a cultural structure more sophisticated than the European and Asiatic Neanderthals and, after a second emigration wave out of Africa, they encountered, clashed against and even mate with their European cousins, some 100,000 years later. It is now widely accepted that, in spite of the fact that both modern humans and Neanderthals had the same cranial capacity the organisational competence of the former displaced, and even caused the latter extinction. It is even possible to speculate that a superior mathematical competence of modern humans contributed to the decisive events that culminated with the extinction of Neanderthals. Strategic thinking involves, among other things, a mathematical sophistication that was probably inferior in Neanderthals when compared with modern humans. In addition, as explained in the previous section, the prefrontal cortex, a well developed area in modern humans and very small in Neanderthals is one of the predominant cerebral area in mathematical processing.

Well before these events, something between 1 and 2 millions ago, *Homo ergaster* started the evolutionary line which culminated with the modern human [8]. Actually, two other species of the genus Homo preceded H. ergaster, namely, *Homo habilis and Homo rudolfensis*. However, the sophistication of the brain which resulted in our cognitive capacity to deal with numbers started probably with *Homo ergaster*. What happened at that time? It is possible that a particular individual happened to be endowed, by a set of more or less complex mutations, with the capacity of numerical processing well above the simple blocking counting. This individual was able to count objects, potential weapons and enemies. His edge over his competitors was obvious. Suppose now that another individual, minimally endowed with the cerebral capacity and with a genecomplex for imitation. If this second individual is able to imitate the counting process adopted by the first one, his edge over competitors increases as well. If the memes for counting are appropriately imitated by those individuals with a higher cerebral capacity, their differential reproductiveness would guarantee that his genes for increased cerebral capacity would start to spread in this population. Hence, the meme-gene-coevolution of mathematical abilities.

7. Future perspectives in the memetics of cognitive mathematics

So far we have addressed the hypothesis of the evolution of our mathematical capacity based on plausible assumptions of meme-gene coevolution of brain size and complexity. It is possible, however, that the evolution of mathematical ability is more dependent on a more active form of meme transmission, namely, teaching. If this is the case then the development of mathematics beyond the subtizing level occurred later in our evolutionary history. Teaching is a more sophisticated form of meme transmission than simple imitation. In addition, teaching is a kind of social interaction which occurs in more complex communities, It may have evolved as a particular case of altruism. Teaching is a form of giving someone else precious informations that could result in differential reproductivity of the receptor individual. Therefore, teaching probably evolved by the same mechanisms currently accepted for the evolution of altruism: kin selection or reciprocation. In the former, the individual that teaches some useful trick to another individual increases his/her inclusive fitness if the receptor shares a substantial number of genes with him/her. In the latter mechanism, the altruistic teacher gives his/her precious information with the expectation that in later interaction the current receptor will share other kinds of information with him/her. Both mechanisms make sense both form the gene and meme points of view. A third mechanism could also be involved in the transmission of mathematical memeplex: selfish teaching. In this form of 'altruism' the teacher shares the information with the receptor basing on the belief that this would increase his/her own survival or the survival of the group. The meme-gene coevolution aspect of this kind of meme transmission, however, should occur under the setting of group selection. Group selection, although still controversial among population geneticists may indeed occur in cultural evolution, as well discussed by Blackmore [2].

The important point we would like to emphasise is that mathematical memeplexes can be transmitted either by imitation or by teaching and that both mechanism can be implied in the evolution of our cognitive mathematical abilities by the same meme-gene coevolutionary mechanisms. Future investigation on these aspects can help clarifying these issues.

In a companion paper we [9] modelled a distributed intelligent system to deal with the process of counting. In this model we showed that it is possible to identify two kinds of processes in the counting mechanism; a linear and very limited process and a oscillatory system which could represent the numerical bases. It is possible that the more primitive (less brained) animals, whose innate counting process is limited to subtizing evolved to our arithmetical capacity by developing the oscillatory system of numerical bases. This is a very rich investigative line that we intend to pursue in future works.

Another area that could be subject of future investigation related to the meme-gene coevolution of mathematical memeplexes would be the development of mathematical tools to deal with the dynamics of meme transmission and the evolutionary aspects, much in the flavour of the neo-Darwinian synthesis. Population genetics is a mathematically rich area of evolutionary biology and perhaps one could propose a new speciality that could be called population memetics in which the adaptive value of meme transmission could be quantified and predictions on meme dynamics could be made. But this also is matter for future works.

8. Final conclusions

Mathematics is a tremendously rich collection of ideas, concepts, tools, etc., that characterises a memeplex. It is always evolving and, in addition to its obvious role in changing the world, and our world vision, had certainly helped in the evolution of our cognitive capacity. Since the first hominids surpassed the imitation threshold, individual mathematical memes more complex than our innate subtizing, in addition to all other memeplexes that characterise human culture, have been transmitted by imitation and later on, by teaching, creating an autocalytic virtuous circle that culminated in the human brain.

References

[1] Dawkins, R., *The Selfish Gene.* Oxford University Press, 1976.

[2] Blackmore, S., *The Meme Machine.* Oxford University Press, 1999.

[3] Dehaene, S., *The Number Sense. How the Mind Creates Mathematics.* Penguin Books, 1997.

[4] Kendal, J. R. and Laland, K. N., Mathematical Models for Memetics. *Journal of Memetics - Evolutionary Models of Information Transmission,* 4. http://jom.emit.cfpm.org/2000/vol4/kendal_jr&laland_kn.html, 2000.

[5] Gabora, L., The Origin and Evolution of Culture and Creativity. *Journal of Memetics - Evolutionary Models of Information Transmission,* 1.http://jom-mit.cfpm.org/vol1/gabora_l.html, 1997.

[6] Blackmore, S. J., Probability misjudgement and belief in paranormal: a newspaper survey. *British Journal of Psychology,* **88,** 1997, pp. 683-689.

[7] Roland, P.E and Friberg, L., Localization of cortical areas activated by thinking. *Journal of Neurophysiology,* **53**, 1985, pp. 1219-1243.

[8] Conroy, G.C., *Reconstructing Human Origins: A Modern Synthesis.* W.W. Norton & Company, 1997.

[9] Rocha, A.F. and Massad, E., Evolving Arithmetical Knowledge in a Distributed Intelligent Processing System, 2002. *This volume.*

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Neuronal Plasticity: How Memes Control Genes

Alfredo Pereira Jr.

Institute of Biosciences, State University of São Paulo (UNESP)
Botucatu - São Paulo, 18618-000, Brazil
apj@ibb.unesp.br

Abstract

This paper gives a outline of the operational steps by which perceived patterns of behavior are transduced from peripheral sensors to the central nervous system, and then to synaptic and intra-neuronal mechanisms called Signal Transduction Pathways (STPs). STPs control cyclic AMP molecules that activate transcription factors leading to differential desinhibition of genes. One of the consequences of the expression of such genes, besides the production of proteins present in the previously activated STPs, is the activation of neural trophic factors (NTFs) that control dendritic growth, and therefore contribute to shape the morphology and patterns of neural connectivity in the brain. As this morphology and dynamical connections influence the behavior of the individual in the social context, a dynamic rapport between cultural patterns ("memes") and his/her genetic system can be established in epigenesis.

Key Words: Plasticity, Neuronal Networks, Transduction, Transcription, Epigenesis.

1. Introduction

The concept of "memes" comes from a darwinian tradition [1] and has contributed to a broader understanding of primate evolution, in terms of a co-evolutionary process involving an interplay of cultural and biological patterns (see Deacon [2]). In this paper I will consider the possibility of studying this interplay from the perspective of cognitive neurobiology.

A meme translates into a complex, usually multimodal sequence of stimuli, generated from a repetitive pattern of behavior in a social context. The perception of a meme involves not only single objects, their features and temporal processes involving them, but also *associations* between such objects, features and processes. For instance, perception of the use of a stone tool involves recognition of different objects and their respective features (tool, hand, substrate modified by the tool, product of the transformation, social consequences of generating this product) and the conception of the process as a whole, including multimodal associations between proprioceptive, tactile and visual information.

The concept of meme also refers to the possibility of cultural reproduction of such behavioral patterns, i.e., the patterns that qualify to be a meme are those susceptible of *being imitated*, thus providing adaptive advantages for the imitators in the group. Therefore, the concept of meme in cognitive neurobiology should also account for a loop between complex perceptual patterns and behavioral schemes ("action schemes", as discussed by Pereira Jr. [3]). Therefore a meme is a perceived behavioral scheme that can be translated by the brain (with the involvement of the pre-motor cortex, as argued by Gallese et al. [4]) into an action scheme, making the imitation of the perceived behavior possible.

A question about the relationship between memes and neuronal plasticity is whether the performance of perception-action loops could shape the brain, both functionally and structurally, along ontogeny. Here I attempt to answer that this influence is possible and also to give a sketch of how it happens.

2. The Genome as a Combinatorial System

The possibility of the brain being shaped by experiences has been clearly defined in recent studies of regulatory processes in the cell. Although all somatic cells in a biological individual share practically the same genes, the patterns of expression of these genes in different tissues is different. In other words, the pool of proteins present in each different tissue is different from the others, although all are produced from the same genes. The solution for this apparent paradox is to conceive the genome as a combinatorial system like natural languages, where a practically infinite number of different sentences can be formed from a finite number of words, which are formed from a relatively small number of letters.

In the genome, the combination of a small number of "letters" (A, T, C, G, U, corresponding to nucleotides adenine, timine, citosine, guanine and uracil) is sufficient to generate different "words" (genes), that can be combined to generate different "sentences" (linear sequences of aminoacids that fold into proteins).

Early in ontogeny, dynamical processes in the cytoplasm, beginning with differential factors already present in the ovule [5], as well as signaling proteins derived from "notch" genes (see Russo [6]), are believed to contribute to embryo differentiation into specialized tissues. Each different tissue has a different pool of proteins, and this difference is maintained since each pool of proteins activate the genome to produce the same proteins (this idea is close to the idea of an "autopoietic" process proposed by Maturana and Varela [7], although these authors didn't focus on the mechanism of tissue differentiation). Such an auto-regulation of protein production is possible because the genome is a combinatorial system susceptible of a large number of different "readings".

Based on this reasoning, Rocha et al. [8] presented a formal grammar (G) composed of a set of symbols corresponding to different nucleotides, and a syntax defined by a set of combinatorial rules:

$$G = \{Vs, Vn, Vt, P, n\}, \qquad (1)$$

where Vs is a set of initial symbols, Vn a set of non-terminal symbols, Vt a set of terminal symbols, n the empty element and P a set of rewriting rules defined as:

$$p\colon a\ Si\ b \dashrightarrow a\ Sj\ b, \qquad (2)$$

where Si is a string of symbols pertaining to the union of all symbol sets, Sj is a string of symbols pertaining to the union of Vs and Vn, and a and b are contextual parameters.

The genetic grammar L is a sub-set of the strings generated by G. Each string accepted as belonging to L is a well-formed formula (*wff*) obtained as the derivation chain (*d*) required to transform an initial symbol into a terminal one:

$$wff(So, Sm) = \mathrm{d}\,(So, Sm) = a\ S1\ b \dashrightarrow \ldots\ a\ Si\ b \dashrightarrow Sm, \qquad (3)$$

where So belongs to Vs and Sm belongs to Vt.

This approach is useful to model gene activation by transcriptional factors and translation of genetic information into proteins. These processes are represented a series of transformations beginning with an initial symbol and ending with a terminal one.

3. Transducing Information From the Environment

Besides the auto-regulation of the pool of proteins determining the differential reading of genes, neuronal tissues have evolutionarily developed a supplementary mechanism to control genetic expression. This mechanism is derived from a specialization of the cellular membrane, forming dendrites, axons and synapses, that allow increased receptivity to informational patterns in the environment, and extended communication between individual neurons, forming integrated networks. Synaptic mechanisms control Signal Transduction Pathways (STPs), in their majority formed by proteins and molecules also present in other kinds of tissues, which transduce synaptic information to the reticulum and nucleus of neurons, thus controlling the dynamics of gene expression along time.

The transduction of information from the environment to the central nervous system (CNS) is made by peripheral sensors that communicate through nerves (parallel bundles of axons) with the CNS. The information being conveyed through this physical channel is encoded in terms of populations of electrical signals. Different hypotheses about such a electrical code have been proposed. Fotheringhame and Young [9] distinguished two classical proposals, the ideas of a population-rate code and of specialized neurons called "feature detectors". These proposals represent two opposite views relatively to the question of *sparseness*: how many neurons are necessary to recognize an aspect of the environment?

The proposal of a population-rate code assumes that information is encoded by the firing *rate* of a spatially distributed population. Studies of transmission from peripheral sensors to the CNS have indicated that this is the kind of coding that operates in the bundles of fibers transmitting afferent information [10, 11].

The feature detector hypothesis, on the other hand, assumes that individual neurons signal exclusively to some *features* of the stimulus, or exclusively to some *kinds* of stimulus. It was inspired by the classical study of the visual system by Hubel and Wiesel [12, 13], when different kinds of cells in the non-striate visual cortex were discovered to fire selectively to specific aspects of the stimulus (e.g., color, boundaries, direction of movement).

Both ideas are not contradictory. They can be combined in the idea of a *sparse population code* [9]. In this view, the detection of real-world objects would be made by a cooperative group of neurons, forming a Hebbian *cell assembly*, a neuronal population with strenghtened connections elicited by previous learning (see e.g. Phillips [14]). A possible solution for difficulties met by both proposals would be to assume that individual neurons participate in different cell assemblies, exhibiting *graded* responses to suboptimal stimuli [9]. These assemblies detect objects as a result of Hebbian (connection-strenghting) learning, when each neuron in the assembly reacts to some component of the pattern elicited by the stimulus.

However, such a combination of encodings may still be not sufficient. Several studies in cognitive neuroscience have suggested that the *temporal* structure of patterns, including the timing of spikes relatively to each other, should be included in models of neuronal computation [15]. In this case new assumptions are necessary:

a) the temporal structure of the evoked response in a cell assembly carries information about the stimulus;

b) the relative timing of spikes in central neuronal assemblies is reliable and also carries information;
c) individual neurons participating in an assembly function as *oscillators*, with less specialization than previously supposed, being able to selectively "resonate" to a range of temporal patterns;
d) the possibility of combination of different temporal patterns across different cell assemblies.

Therefore, the logical grammar of the electrical neural code should account for three kinds of factors:

a) spatially distributed frequencies of action potentials, as measured by the EEG;
b) differential responses of specialized agents, as measured by single unit electrodes, and
c) temporal structure of spike patterns, also measured by single units and analyzed with a variety of methods [16, 17].

4. Transducing Information at the Synapse

When electrical pulses reach the synaptic terminal and open vesicles, thus releasing transmitters in the synaptic cleft, the patterns of information encoded in electromagnetic form is transduced to a chemical/molecular code. This code is composed of three kinds of elements: transmitters, receptors and neuromodulators. Such elements have affinity or not to each other. When they combine (combinations of two, three or more elements) they originate specialized *agents*. These agents can be classified in five categories:

a) membrane excitatory agents: these agents are composed by associations of a transmitter (Acetylcholine, AMPA, Kainate) with an ionotropic receptor (a receptor that control the flux of ions - Na, K, and Cl ions - as the nicotinic and muscarinic acetylcholine ones), acting to depolarize the neuronal membrane. When such depolarization reaches a threshold the neuron fires and then converts the information back to electrical coding. This association can be controlled by a third element (a ligand), in an "allosteric" fashion [18];
b) membrane inhibitory agents: composed by associations of transmitters (GABA) with an ionotropic receptor acting to hyperpolarize the neuronal membrane;
c) controlling calcion ions entrance into the post-synaptic neuron: glutamate transmitter binding to the NMDA (N-methyl-d- aspartate) receptor controls the entrance of calcium ions, that trigger a diversity of metabolic activities in the cytoplasm, including facilitation of membrane electric activity by means of retrograde messengers Nitric Oxide and Arachidonic Acid, and activation of transcriptional factors;
d) unspecific sensitization: monoamine transmitters (Serotonin, Dopamine) binding with metabotropic receptors increase the sensitivity of signal transduction pathways (STPs) to a wide range of metabolic processes, including facilitation of electric activity and activation of transcriptional factors;
e) specific sensitization: binding of monoamine transmitters with metabotropic receptors coupled to a G-protein (Guanidine Triphosphate-binding family of proteins) modulated by a function-specific neuromodulator (as neuropeptides; see Merighi [19] for the coexistence of neuropeptides with a variety of receptors) leads to the activation of a specific STP.

Therefore the binding of monoamines with metabotropic receptors coupled to G-proteins, as well as the entrance of Ca ions through NMDA channels transduce electrically-encoded information into intra-cellular STPs, which are able to reach the nucleus and control the expression of genes.

5. Transducing Information From Synapse to Nucleus

Rocha [20] distinguished three kinds of STPs, those regulated by G-proteins, Ca entry and also PTK (Protein Tysosine Kinase) pathways. The latter controls growth factors that act from the inside of the neuronal membrane.

Rocha's symbolic representation of the molecular syntax of STPs refers to a set of transmitters (T), responsible to trigger a STP by binding with a membrane receptor (R), that may be the NMDA receptor or a metabotropic one, according to the above section. R binds to a secondary messenger (M), that activate a chemical chain that controls gene reading. This is the case when the signal transduction chain involves transcriptional factors (F) activated by cyclic Adenosine Monophosphate (cAMP).

Therefore, a STP grammar (STP) can be represented by:

$$\mathrm{STP} = \{Vt,\ Vr,\ Vm,\ Vf,\ P,\ n\}, \qquad (4)$$

where Vt is the set of initial symbols belonging to T, Vr the set of intermediary symbols belonging to R, Vm the set of intermediary symbols belonging to M, Vf the set of terminal symbols belonging to F, and P and n are defined as in (2).

Strings generated by the STP grammar are represented as a well-formed formula (*wff*) obtained as the derivation chain (*d*) of symbols S required to transform an initial symbol into a terminal one:

$$wff(So, Sr, Sm, Sf) = \mathrm{d}\,(So, Sr) = a\ S1\ b \dashrightarrow \ldots\ a\ Si\ b \dashrightarrow Sf \qquad (5)$$

where So belongs to Vt, Sr belongs to Vr, Sm belongs to Vm and Sf belongs to Vf.

When the action of neuromodulators is also considered, the initial step of the processing is then considered as a triplet of transmitter, receptor and modulator. For the sake of simplicity, this possibility was not considered in (5).

A STP string ends with a symbol that binds to an initial symbol belonging to the genetic language, then triggering gene reading and protein production. Such a protein is an initial symbol or an effector that activates an initial symbol of the STP grammar. This kind of working of cells enable them to self-organize, allowing environmental influences to be absorbed, and also engendering coherent adaptive action of the organism in the same environment.

Computer simulations of the interaction of different STPs have been performed [21]. They reveal complex regulatory loops, pathways that lead to gene expression producing proteins that replace proteins necessary for the continuity of the same processes, as well as growth factors that control the Hebbian strenghtning of inter-neuronal connections supporting learning and memory. These connections allow the creation of new and recombination of previously existing action schemes, supporting a variety of social behaviors, which may reinforce the same memes that stimulated the creation and recombination of such schemes.

6. Acknowledgements

I would like to express my gratitude to Drs. Armando F. Rocha and Eduardo Massad for their encouragement to write this paper, and to CNPQ/Brasil for financial support.

References

[1] DAWKINS, R., *The Selfish Gene*. Oxford University Press, 1990.
[2] DEACON, T.W., *The Symbolic Species: The Co-Evolution of Language and the Brain*. W.W. Norton and Co., 1997.
[3] PEREIRA JR., A., A Possible Role for Action Schemes in Mirror Self-Recognition. *Revista de Etologia* **2**, 1999, pp. 127-139.
[4] GALLESE, V., FADIGA, L., FOGASSI, L. and RIZZOLATI, G., Action Recognition in the Premotor Cortex. *Brain* **119**, 1996, pp. 593-609.
[5] PEREIRA JR., A.., GUIMARÃES, R.. e CHAVES JR., J. C., Auto-Organização na Biologia: Nível Ontogenético. In: Debrun, M., Gonzales, M.E. e Pessoa, O. (Eds.) *Auto Organização - Estudos Interdisciplinares*. Centro de Lógica e Epistemologia/ UNICAMP, 1996.
[6] RUSSO, E., Interpreting the Signaling of Notch: Delineation of Biochemical Pathway Points to Possible Alzheimer's Therapy. *The Scientist* **15** (4), 2001, pp. 19-22.
[7] MATURANA, H. and VARELA, F., *Autopoiesis and Cognition: the Realization of the Living*. Boston Studies in the Philosophy of Science 42/Reidel, 1980.
[8] ROCHA, A.F., REBELLO, M.P. and MIURA, K., Toward a Theory of Molecular Computing. *Information Sciences* **106** (1/2), 1998, pp. 123-157.
[9] FOTHERINGHAME, D.K. and YOUNG, M.P., Neural Coding Schemes for Sensory Representation: Theoretical Proposals and Empirical Evidence. In Rugg, M.D(Ed.) Cognitive Neuroscience. Cambridge, MIT Press, 1997.
[10] CONNOR, C.E., HSIAO, S.S., PHILLIPS, J.R. and JOHNSON, K.O., Tactile Roughness: Neural Codes That Account for Psychophysical Magnitude Estimates. *The Journal of Neuroscience* **10** (12), 1990, pp. 3823-3836.
[11] CONNOR, C.E. and JOHNSON, K.O., Neural Coding of Tactile Texture: Comparison of Spatial and Temporal Mechanisms for Roughness Perception. *The Journal of Neuroscience* **12** (9), 1992, pp. 3414-3426.
[12] HUBEL, D.N. & WIESEL, T.N., Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex. *Journal of Physiology* **160**, 1962, pp.106-154.
[13] HUBEL, D.N. & WIESEL, T.N., Receptive Fields and Functional Architecture of Monkey Striate Cortex. *Journal of Physiology* **195**, 1968, pp. 215-243.
[14] PHILLIPS, W.A. (1997) Theories of Cortical Computation. In Rugg, M.A. (Ed.) Cognitive Neuroscience. Cambridge: MIT Press, 1997.
[15] CARIANI, P., As Time Really Mattered: Temporal Strategies for Neural Coding of Sensory Information. In Pribram, K.(Ed.) *Origins: Brain and Self-Organization*, Hillsdale: Erlbaum, 1994.
[16] HOPFIELD, J.J., Pattern Recognition Computation Using Action Potential Timing for Stimulus Representation. *Nature* 376, 1995, pp. 33-36.
[17] RIEKE, F., WARLAND,D., STEVENINCK, R.R. and BIALEK, W., *Spikes*. Cambridge: MIT Press, 1997.
[18] CHANGEUX, J.P. and EDELSTEIN, S.J., Allosteric Mechanisms in Normal and Pathological Nicotinic Acetylcholine Receptors. *Current Opinion in Neurobiology* **11** (3), 2001, pp. 369-377.
[19] MERIGHI, A., Costorage and Coexistence of Neuropeptides in the Mammalian CNS. *Progress in Neurobiology* **66** (3), 2002, pp. 161-190.
[20] ROCHA, A.F., The Brain as a Symbol-Processing Machine. *Progress in Neurobiology* **53**, 1997, pp. 121-198.
[21] BHALLA, U.S. and Iyengar, R., Emergent Properties of Networks of Biological Signaling Pathways. *Science* **283**, 1999, pp. 381-387.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

The influence of heterogeneity in the control of diseases

Laécio C. de Barros[1] Rodney C. Bassanezi[1]
Renata Zotin G. de Oliveira[2]

[1] DMA - IMECC - UNICAMP, 13.081-970 - Campinas/SP, Brazil
[2] IGCE- UNESP, 13.506-700 - Rio Claro/SP, Brazil

Abstract. In this paper we consider the SIS epidemiological model (susceptible-infected-susceptible) in which the transmission and recuperation rates are considered fuzzy sets. A comparative study of the equilibrium points of the disease for the classical and fuzzy models is done. In addition, a program of disease control is suggested based on the Basic Reproduction Value, R_0^f, which agrees with the one adopted by experts in the public health field.

Keywords. Fuzzy Sets, Epidemiology, Basic Reproduction Value, Fuzzy Expected Value.

1 Introduction

The models of Kermack and McKendric are the first mathematical models in Epidemiology. These models consider that all the infected individuals have the same chance of infecting susceptible in each meeting. More recent models consider different factors that influence the occurence of a new infection, among them, the viral charge of infected individuals. Aiming to obtain information about disease control and to do more precise analysis, it is common to consider the infected class subdividing it into n different stages, according to the infectivity degree of the individuals. In this case, the complexity of the model increases, making difficult the analysis of some epidemiological parameters, like the *Basic Reproduction Value* R_0 [4].

The mathematical treatment of gradual uncertainties, like one used to differentiate the individuals within a population, has utilized more and more techniques of Fuzzy Theory. Sadegh-Zadeh [7] distinguishes the individuals according to their health state, considering intermediary stages between health and disease.

In [2] a model is proposed in which the heterogeneity of infected population is taken into account, and in which the individuals infect differently according to their viral charge. For this, the contact rate was considered as a fuzzy set, and it was then possible to obtain a Basic Reproduction Value R_0^f which is different from the deterministic model, R_0.

In this paper we present more information about disease control based on R_0^f. In addition, a comparative study between the equilibrium points of the disease for the the classical and fuzzy SIS models is done.

2 Preliminaries

A fuzzy subset F of the universe set U is simbolically represented by the membership function $u : \mathcal{U} \rightarrow [0,1]$, where $u(x)$ indicates the degree of membership of x in the fuzzy set F.

In order to use a defuzzification method, we need the concept of fuzzy measure. Let Ω be a non-empty set and $P(\Omega)$ the power sets of Ω.

The function $\mu : P(\Omega) \rightarrow [0,1]$ is a fuzzy measure if

a) $\mu(\emptyset) = 0$ and $\mu(\Omega) = 1$ b) $\mu(A) \leq \mu(B)$ if $A \subseteq B$.

As a defuzzifier, we are going to use the fuzzy integral or *Fuzzy Expected Value* of fuzzy set that is given by

$$FEV[u] = \sup_{0 \leq \alpha \leq 1} \inf[\alpha, H(\alpha)],$$

where $H(\alpha) = \mu\{x \in : \Omega : u(x) \geq \alpha\}$. Thus, $H(\alpha)$ is a decreasing function and the $FEV[u]$ is the fixed point of $H(\alpha)$.

Sugeno [6] has proved that $\mid FEV[u] - E[u] \mid \leq 0.25$ if μ is a probability measure, u is both a fuzzy set and random variable and $E[u]$ the classical expectancy of u.

3 The Model

The simplest mathematical model to describe the dynamics of directly transmited diseases with interaction between susceptible and infected individuals, which do not confer immunity, is the SIS model without vital dynamics. Mathematicaly, it is described by a non-linear system of differential equations

$$\begin{cases} \frac{dS}{dt} &= -\beta SI + \gamma I \\ \frac{dI}{dt} &= \beta SI - \gamma I \end{cases} \tag{1}$$

where S is the proportion of susceptible individuals, I is the proportion of infected individuals at each instant, β is the transmission coefficient of the disease and γ is the recuperation coefficient of the infectious individuals.

We assume that the heterogeneity of a population is just given by the infected individual's viral charge (see the fuzzy notion of disease[7]). Thus, the higher the viral-charge, the higher will be the chance of disease transmission. In other words, we assume that $\beta = \beta(v)$ measures the chance of transmission occuring in a meeting between a susceptible and an infected individual with an amount of virus v.

To obtain the membership function of β, we assume that, when the amount of virus in an individual is relatively low, the chance of transmission is negligible, and that there is a minimum amount of virus $v_{\min}$ needed to cause disease transmission. This value $v_{\min}$ can be understood as the one which gives the susceptibility of a particular group. In fact, the higher the $v_{\min}$value, the higher the amount of virus needed for transmission to occur and it means that the group has a low susceptibility to the disease.

Furthermore, from a certain amount of virus v_M, the chance of disease transmission is maximum and equal to 1. Finally, we suppose that the individual's amount of virus is always limited by $v_{\max}$ for each disease. Likewise, the recuperation coefficient $\gamma = \gamma(v)$ should be a decreasing function of v,with a minimun value $\gamma_0 > 0$.

We have chosen for the fuzzy subsets β and γ, the following membership functions ([1]):

$$\beta(v) = \begin{cases} 0 & \text{if} \quad v \notin [v_{\min}, v_{\max}) \\ \dfrac{v - v_{\min}}{v_M - v_{\min}} & \text{if} \quad v \in (v_{\min}, v_M] \\ 1 & \text{if} \quad v \in (v_M, v_{\max}] \end{cases} \quad \text{and} \quad \gamma(v) = \frac{(\gamma_0 - 1)}{v_{\max}} v + 1$$

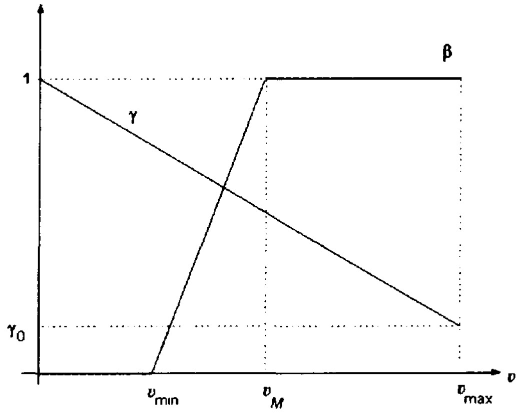


Figure 1: $\beta(v)$ and $\gamma(v)$

4 Fuzzy Basic Reproduction Value (R_0^f)

Traditionaly, a disease control program is based on the Basic Reproduction Value R_0, which gives the number of secondary cases caused by an infected individual introduced into a whole susceptible population.

For the classical model SIS we have $R_0 = \frac{\beta}{\gamma}$ and the disease will be extinct if $R_0 < 1$, and it will become established in the population if $R_0 > 1$ [3].

The use of $R_0(v)$ to control the disease pressuposes the knowledge of the viral charge to the whole population. To make the model more realistic, we consider that the viral charge $v \in [0, v_{\max}]$ has different chances of occurence in the population, i.e., each individual contributes differently to the disease propagation, whose membership function is given by $\rho(v)$

$$\rho(v) = \begin{cases} 0 & \text{if} \quad v \notin [\bar{v} - \delta, \bar{v} + \delta] \\ \frac{1}{\delta}(v - \bar{v} + \delta) & \text{if} \quad v \in [\bar{v} - \delta, \bar{v}] \\ -\frac{1}{\delta}(v - \bar{v} - \delta) & \text{if} \quad v \in (\bar{v}, \bar{v} + \delta] \end{cases}$$

The parameter $\bar{v}$ is the median viral charge (in this case, the median viral charge of the infected individuals), and δ is the dispersion. Observe that $\rho(v)$ is typically a membership function of a triangular fuzzy number [5].

The Fuzzy Basic Reproduction Value (R_0^f) will be defined using the concept of Fuzzy Expected Value (FEV). This new parameter is a kind of average of $R_0(v)$, taking into account the distribution of the viral charge v, according to the membership function $\rho(v)$.

Note that $R_0(v)$ isn't a fuzzy set since it can be bigger than one. However, it is easy to see that the maximum value of $R_0(v)$ is $\frac{1}{\gamma_0}$. Then, $\gamma_0 R_0(v) \leq 1$, indicating that $\gamma_0 R_0$ is a fuzzy set and in this case $FEV[\gamma_0 R_0]$ is well-defined.

We define the Fuzzy Basic Reproduction Value for the whole population by:

$$R_0^f = \frac{1}{\gamma_0} FEV[\gamma_0 R_0]. \tag{2}$$

As seen in the preliminary section,

$$FEV[\gamma_0 R_0] = \sup_{0 \leq \alpha \leq 1} \inf[\alpha, H(\alpha)]$$

where $H(\alpha) = \mu\{v : \gamma_0 R_0(v) \geq \alpha\} = \mu(X)$ and μ is a fuzzy measure.

The following possibility measure will be used as a fuzzy measure [5]:

$$\mu(A) = \sup_{v \in A} \rho(v).$$

This one can be considered a very reasonable measure to be adopted by experts who do not want to take risks in their evaluations and possible treatments, because it is quite a conservative measure in the sense that the infectivity degree of a group is represented by the individual with the highest degree of infectivity.

Returning to the calculation of $FEV[\gamma_0 R_0]$, observe that as $\frac{\beta(v)}{\gamma(v)}$ is not decreasing with v. Therefore, the set X is an interval of the form $[v', v_{\max}]$ where v' is the solution to the equation

$$\gamma_0 \frac{\beta(v)}{\gamma(v)} = \alpha.$$

Then,

$$H(\alpha) = \mu[v', v_{\max}] = \sup_{v' \leq v \leq v_{\max}} \rho(v). \tag{3}$$

Observe at first, that $H(0) = 1$ and $H(1) = \rho(v_{\max})$. On the other hand, if $\alpha > 0$ and since $\frac{\beta(v)}{\gamma(v)}$ is increasing, we have

$$v' = \frac{[\alpha(v_M - v_{\min}) + \gamma_0 v_{\min}] v_{\max}}{\gamma_0 v_{\max} - \alpha(\gamma_0 - 1)(v_M - v_{\min})} \qquad \text{if} \qquad 0 < \alpha < \frac{\gamma_0}{\gamma(v_M)}$$

or

$$v' = \frac{v_{\max}}{\gamma_0 - 1} \left(\frac{\gamma_0}{\alpha} - 1 \right) \qquad \text{if} \qquad \frac{\gamma_0}{\gamma(v_M)} \leq \alpha < 1$$

In order to exemplify the evaluation of $FEV[\gamma_0 R_0]$, we will assume that the viral charge V of a group of individuals is a linguistic that assumes the classifications: weak (V_-), strong (V^+) or medium (V_-^+). Each classification is a fuzzy number based on the values $v_{\min}$, v_M and $v_{\max}$ that appear in the definition of β (see the Figure 2).

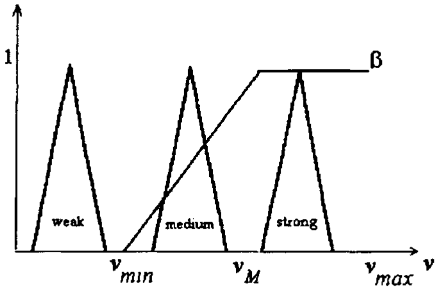


Figure 2: Classification of Linguistic Variable (V), according to $\rho(v)$

- Case a) Weak viral charge (V_-), which is characterized by $\bar{v} + \delta < v_{\min}$. Since $\bar{v} + \delta < v'$ then

  $$H(\alpha) = \sup_{v' \le v \le v_{\max}} \rho(v) = 0, \qquad \forall \alpha \in [0, 1].$$

  and

  $$FEV[\gamma_0 R_0] = 0 < \gamma_0 \Longleftrightarrow R_0^f < 1$$

  which makes it possible to conclude that the disease will be extinct.

- Case b) Strong viral charge (V^+), which is characterized by $\bar{v} - \delta > v_M$ and $\bar{v} + \delta < v_{\max}$. Then, using (3), we have:

  $$H(\alpha) = \begin{cases} 1 & \text{if} \quad 0 \le \alpha < \dfrac{\gamma_0}{\gamma(\bar{v})} \\ \rho(v') & \text{if} \quad \dfrac{\gamma_0}{\gamma(\bar{v})} \le \alpha < \dfrac{\gamma_0}{\gamma(\bar{v} + \delta)} \\ 0 & \text{if} \quad \dfrac{\gamma_0}{\gamma(\bar{v} + \delta)} \le \alpha \le 1 \end{cases}$$

  The function H is decreasing, and it is easy to see if $\delta > 0$, H is continuous. Then, $FEV[\gamma_0 R_0]$ is the fixed point of H and we conclude that

  $$\frac{\gamma_0}{\gamma(\bar{v})} < FEV[\gamma_0 R_0] < \frac{\gamma_0}{\gamma(\bar{v} + \delta)}.$$

  Therefore, as $\frac{\gamma_0}{\gamma(\bar{v})} > \gamma_0$, it follows that $R_0^f > 1$, which indicates that the disease will become established in the studied group.

- Case c) Medium viral charge (V_-^+), which is characterized by $\bar{v} - \delta > v_{\min}$ and $\bar{v} + \delta < v_M$.

Thus, from (3), we have:

$$H(\alpha) = \begin{cases} 1 & \text{if } 0 < \alpha \leq \gamma_0 \dfrac{\beta(\overline{v})}{\gamma(\overline{v})} \\ \rho(v') & \text{if } \gamma_0 \dfrac{\beta(\overline{v})}{\gamma(\overline{v})} < \alpha \leq \gamma_0 \dfrac{\beta(\overline{v}+\delta)}{\gamma(\overline{v}+\delta)} \\ 0 & \text{if } \gamma_0 \dfrac{\beta(\overline{v}+\delta)}{\gamma(\overline{v}+\delta)} < \alpha \leq 1 \end{cases}$$

As in case b), we have

$$\frac{\beta(\overline{v})}{\gamma(\overline{v})} < R_0^f < \frac{\beta(\overline{v}+\delta)}{\gamma(\overline{v}+\delta)}.$$

Next, we will study the equilibrium points of the fuzzy model.

5 Equilibrium points: deterministic versus fuzzy

Considering the viral charge present in the population, from equations system (1) we have the following equilibrium points:

$$P_1 = (1,0) \text{ and } P_2(v) = (S_2(v), I_2(v)) = (\frac{\gamma(v)}{\beta(v)}, 1 - \frac{\gamma(v)}{\beta(v)})$$

A stability analysis of the classical model (1) shows that P_1 is unstable while P_2 (with $\beta \geq \gamma$) is asymptoticaly stable. Observe that v^* for which $\beta(v^*) = \gamma(v^*)$ is the bifurcation value.

Since the viral charge has different possibilities of occurence, we can say the same for the equilibrium points $P_2(v)$. Then, we calculate the average number for these points $(\bar{P}_2)$ and compare with the deterministic case $P_2(\overline{v})$ (see, the Figure 3).

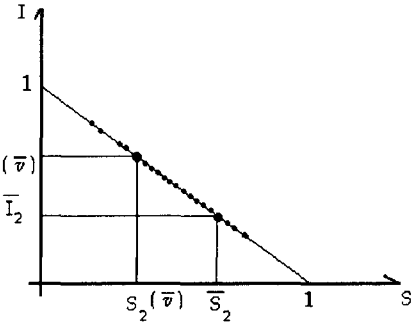


Figure 3: Equilibrium points for the deterministic and fuzzy models

Let $\bar{P}_2 = P_2(\bar{S}_2, \overline{I}_2) = P_2(FEV(S_2(v)), FEV(I_2(v)))$.

Since $S_2(v) = 1/R_0(v)$, we have $\bar{S}_2 \geq S_2(\bar{v})$, because

$$S_2(\bar{v}) = \frac{1}{R_0(\bar{v})} \leq \mu\{v : \frac{1}{R_0(v)} \geq \frac{1}{R_0(\bar{v})}\} = \mu\{v : R_0(v) \leq R_0(\bar{v})\}$$

Consequently, $\bar{I}_2 < I_2(\bar{v})$, that is, the proportion of infected individuals, considering all the viral charges, is less than the proportion of infected individuals when only the median viral charge is considered (assuming population homogeneity).

6 R_0^f as a parameter for control and conclusions

For the classification of viral charge used earlier, it is possible to obtain precise values for R_0^f :

- for the weak viral charge, one quickly concludes that $R_0^f = 0$;
- for the strong viral charge, the value of $FEV[\gamma_0 R_0]$ is the fixed point of H which is given by the solution to the equation:

  $$\rho(v') = \alpha \Leftrightarrow \frac{\bar{v} - v'}{\delta} + 1 = \alpha$$

  with $v_M < \bar{v} \leq v' \leq v_{\max}$.

  Thus, using the expression v' for this case, $FEV[\gamma_0 R_0]$ is the only positive solution to the second degree equation:

  $$(1 - \gamma_0)\delta\alpha^2 + [(\gamma_0 - 1)(\bar{v} + \delta) + v_{\max}]\alpha - \gamma_0 v_{\max} = 0.$$

- for the medium viral charge, we have $v_{\min} \leq v' \leq v_M$. Then, $FEV[\gamma_0 R_0]$ is the only positive solution to the second degree equation:

  $$(\gamma_0 - 1)\delta\alpha^2 - [(\gamma_0 - 1)(\bar{v} + \delta) + v_{\max} - \frac{\gamma_0 \delta v_{\max}}{v_M - v_{\min}}]\alpha + \gamma_0 v_{\max}\beta(\bar{v} + \delta) = 0. \quad (4)$$

  In either one of the cases, from section 4, we have

  $$R_0(\bar{v}) < R_0^f < R_0(\bar{v} + \delta).$$

As the function $R_0(v) = \frac{\beta(v)}{\gamma(v)}$ is increasing and continuous, according to the Intermediate Value Theorem, there exists only one $\hat{v} \in (\bar{v}, \bar{v} + \delta)$ such that

$$R_0^f = \frac{\beta(\hat{v})}{\gamma(\hat{v})} = R_0(\hat{v}) > R_0(\bar{v}), \quad (5)$$

that is, there exists only one viral charge $\hat{v}$ such that R_0 (classical) and R_0^f (fuzzy) coincide. In addition, the average number of secondary cases (R_0^f) is higher than the number of secondary cases ($R_0(\bar{v})$) due to the median viral charge.

To reinforce the use of system (1) with the parameter $\hat{v}$, in the sense of describing the dynamic of the disease in the whole population, we will analyze the control of the evaluation of the disease in the population using $R_0(\hat{v}) = R_0^f$:

- for the case in which the viral charge is weak, we have $\hat{v} < \overline{v}+\delta \leq v_{\min}$. Then $R_0(\hat{v}) = 0$ and the disease will not be established in the population.
- for the case in which the viral charge is strong, we have $\hat{v} > \overline{v} > \overline{v}+\delta \geq v_M$. Thus, $R_0(\hat{v}) = \frac{1}{\gamma(\hat{v})} > 1$, indicating that the disease will become established.
- for the case in which the viral charge is medium, we have:
  - if $v^* > \hat{v}$ então $R_0(\hat{v}) = \dfrac{\beta(\hat{v})}{\gamma(\hat{v})} < \dfrac{\beta(v^*)}{y(v^*)} = 1$,indicating the disease will be extinct.
  - if $v^* < \hat{v}$ então $R_0(\hat{v}) = \dfrac{\beta(\hat{v})}{\gamma(\hat{v})} > \dfrac{\beta(v^*)}{y(v^*)} = 1$, indicating the disease will be present in the population, where v^* is the bifurcation value, as seen in the section 5.

Now, the adoption of R_0^f agrees with the policy of disease control used by public health experts:

1. Since R_0^f is obtained as a positive solution of second degree equation, it is not difficult to see that its reduction can be obtained by increasing $v_{\min}$ (consequently increasing v^*), or in other words, by decreasing the susceptility of the group. This can be accomplished by improving the quality of life of the study population.
2. Since $\hat{v} \in (\overline{v}, \overline{v}+\delta)$, R_0^f can be diminished by decreasing the median viral charge; for example, through the use of medicine or isolation of infected individuals (decreasing δ).

In this study we proposed a possible way to use Fuzzy Theory to model the uncertainty present in Epidemiology. Our main conclusions are that, if the uncertainty is excluded before modelling (e. g. a deterministic model), important parameters of management and control of phenomenon are not being well evaluated. Moreover, it can occur that each parameter is an underestimate of the real one, as was seen above: $R_0(\overline{v}) < R_0^f$.

Acknowledgments: This research was supported by CNPq and FUNDUNESP.

References

[1] L.C.Barros, R.C.Bassanezi and M.B.F.Leite, The epidemiological models SI with fuzzy parameter of transmission. (submitted).

[2] L.C.Barros, R.C.Bassanezi, R.Z.G.Oliveira and M.B.F.Leite, A disease evolution model with uncertain parameters. Proceedings of 9th IFSA World Congress and 29th NAFIPS International Conference, Vancouver, (2001)

[3] L. Edelstein-Keshet, Mathematical Models in Biology. In: **Birkhäuser Mathematics Series**. McGraw-Hill Inc., New York, (1988).

[4] M.B.F.Leite, R.C.Bassanezi and H.M.Yang, The basic reprodution for a model of directly transmitted infections considering the virus charge and the immunological response ". *IMA Journal of Mathematics Applied in Medicine an Biology* **17** (2000) 15-31.

[5] H.T.Nguyen and E.A.Walker, *A First Course in Fuzzy Logic*, CRC Press, Inc. (1997).

[6] M.Sugeno, *Theory of Fuzzy Integrals and Its Applications*. Doctoral Thesis, Tokyo Institute of Technology. (1974).

[7] K.Sadegh-Zadeh, Fundamentals of clinical methodology: 3. Nosology - *Artificial Intelligence in Medicine* 17 (1999) 87-108.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Paraconsistent Logics viewed as a Foundation of Data Warehouses

Seiki Akama
Computational Logic Laboratory, Department of Information Systems,
Teikyo Heisei University, 2289 Uruido, Ichihara-shi,
Chiba, 290-0193, Japan.
e-mail akama@thu.ac.jp

Jair Minoro Abe
Department of Informatics, ICET, Paulista University,
R. Dr. Bacelar 1212, 04026-002, Sao Paulo, SP, Brazil.
e-mail jairabe@uol.com.br

Abstract
Data warehouse can be seen as a large database in which we can use several techniques from data mining and knowledge discovery in database (KDD). Data warehouses store many data which may have features related to incomplteness and inconsistency. To provide a foundation of data warehouses, paraconsistent logics are useful. In this paper, we sketch an outline of data warehouses using existing systems of paraconsistent logics. We compare these systems in several respects and claim that most systems are suitable to KDD. We also argue that some extensions of paraconsistent logics can be developed to show the mechanism of data mining.

Keywords
Paraconsistent logics, data warehouse, database, data mining, knowledge discovery in database (KDD).

1 Introduction

Database now becomes one of the necessary software systems in our daily life. As a consequence of the development of memory technology, it is possible to store a large number of data in a database. Database theory can serve as a basis of efficient storage and data retrieval from large databases, in particular, *relational databases*. Recently, large databases are attracted under the name of *data warehouses* since the proposal of Inmon (1992). We now expect to see the promising ways of abilities to analyze, summarize, and extract meaningful knowledge from data in data warehouses. Thus, the area called *data mining* emerged. Data mining aims at discovering unexpected patterns and new rules hidden in large databases like data warehouses. In this sense, data mining is regarded as part of more sophisticated topic known as *knowledge discovery in database* (KDD). In other words, KDD is the general process of discovering useful knowldge from data by means of *some* techniques in data mining; see Adrians and Zantinge (1996).

As is obvious from the literature on data mining, the researchers seem to employ methods from several fields such as statistics, fuzzy theory, neural network. From a theoretical point of view, it is interesting to work out logical foundations of data warehouses and KDD. This means that processes in KDD are interpreted in a logcal setting if data warehouses are formalized as logic databases. Unfourtunately, such an attemt has not been investigated in databababse community. The purpose of this paper is to propound a logic-based foundation of data warehouses. The novel feature of the present work is found in the use of the so-called *paraconsistent logics* rather than classical logic.

The plan of the paper is as follows. In section 2, we review data warehouses in connection with databases. In section 3, we discuss paraconsistent logics by pointing out several features. In section 4, we propose to view paraconsistent logic databases as data warehouses. From the interpretation, data warehouses are possibly incomplete and inconsistent logic databases. In section 5, we further argue that possible extensions of paraconsistent logics can provide the techniques in KDD. Section 6 concludes the paper with the discussion on future work.

2 Data Warehouses

By *data warehouses*, we mean a large database that is a basis of extracting "hidden" information of data. Here, such informaion is available by methods in *data mining*. And many techniques of data mining are gathered and discussed in the area of *knowdge discovery in database* (KDD). However, the idea of data warehouses is not new and redicovered in the literature on computer science. One of such topics is undoubtedly found in the work on *decision support system* (DSS) and *management information system* (MIS); see Date (2000). Roughly speaking, these systems were skecthed as intelligent systems to help our decision and for this purpose we need some kind of "database".

Below we use the term "data warehouse" to mean a class of such intelligent systems. In this regard a data warehouse is a database with the following mechanisms:

- data input
- information retrieval
- data manipulation
- modelling

- analysis
- reporting

Data input is a process of constructing a database. *Information retrieval* is to extract the relevant information from data. For instance, in relational database, it can be done by SQL. *Data manipulation* is to transform data suitable to the analysis. *Modelling* is to specify a model of decision support. Based on the given model, *analysis* gives data an interpretation by some method from fields mainly in statistics. After the analysis, *reporting* outputs the result in a document. One can see that the recent database systems also have similar mechanisms. In this sense, the point is that data warehouses should be interpreted as "intelligent" databases in which the relevant information can be extracted.

Investigators in the area agree that data warehouses should have the following features. First, they are subject-oriented in that data depend on a certain subject. Second, they are integrated in that data in data warehouses should also satisfy integrity constraints. Third, they are time-variant in that the input time of data is crucial. Fourth, they are non-volatile in that no data is updated.

These features lead us to consider a foundation of data warehouses different from that of databases. Here are some important points. A database is usually *consistent* in the sense of some data model. On the other hand, data warehouses allow to store large data which may inconsistent each other following the above mentioned features. By non-volatility, new data may contradict old data. But, by integrity, we need possible constraints in inconsistent data warehouses. In addition, the subject-orientedness means that data belong to some domain in discourse. And we should integrate the notion of time due to the time-variance.

If we consider a logic-based formalization of data warehouses, it must satisfy these features. But, in this paper, we starts with the foundation satisfying integrity and non-volatility. The remaining features will be reconsidered in possible extensions of the foundation suggested in section 5.

3 Paraconsistent Logics

If we model a database in logic, several notions like data model, integrity constraint, query should be logically formalized. This is a starting point of *logic database*; see Gallair and Minker (1976). One might pursue a foundation of data warehouses in this line. However, this line of extension is problematic. The reason is that "logic" used in logic database is *classical logic*. The difficulty lies in the inability of handling inconsistency of classical logic. Since classical logic is consistent in the sense that both A and $\neg A$ hold, it cannot satisfy non-volatility. If we have a contradiction in classical logic, we can deduce arbitrary conclusions. This means that we cannot extract relevant information from the inconsistent database. Therefore, we can conclude that the logic database based on classical logic has no extensions relevant to data warehouses.

Fourtunately, there are logical systems capable of fomalizing reasoning with inconsistency. In the literature, such systems are known as *paraconsistent logic*.

Here, we discuss paraconsistent logic more formally. Let Γ be a set of formulas and A a formula. We write $\Gamma \vdash_L A$ to mean that A is provable from Γ in a logic L. If Γ is empty set, i.e. $\vdash_L A$, the notation says that A is a theorem in L. We say that L is *inconsistent* if $\vdash_L A \& \neg A$, and *consistent* otherwise. Say that L is *trivial* if $\vdash_L B$ for any B. In classical logic, inconsistency induces triviality, namely $A \& \neg A \vdash_L B$ holds. But, the feature is undesirable for many applications and is a limitation of clasical logic.

We can now classify a paraconsistent logic as a logic for inconsistent but non-trivial theories. There are a number of systems of paraconsistent logics in the literature. We can list the following three main approaches:

- discussive logic
- da Costa logic
- relevance logic

Discussive logic was proposed by Jaskowski (1948) to formalize discourse in which each participant gives some information, belief, and so on. In general, such information supplied by different participants may contradict each other. To advance a contradictory system, in discussive logic, the inference from A and B to $A \& B$, called adjunction, fails. For this reason, discussive logic is classified *non-adjunctive systems*.

Da Costa logic was invented by da Costa (1974) to develop a logic for inconsistent form systems, which belongs to the so-called C-systems. The gist of da Costa logic lies in the presence of non-truth-functional negation in positive logic. By this, da Costa logic succeeds in blocking the law of non-contradiction $\neg(A \& \neg A)$.

Relevance logic was a family of logics proposed by Anderson and Belnap (1975) aiming at developing the logic of relevant implication. In connection with this attempt, most relevance logics have the negation in paraconsistent flavour in that both A and $\neg A$ could be true.

All of these three systems are basically paraconsistent in the above sense. Of course, other versions of paraconsistent logics have been proposed in the literature. For example, *many-valued logics* are closely related to paraconsistent logics; see Belnap (1977). Belnap's four-valued logic has four truth-values, i.e. *true*, *false*, *none*, *both*. Here, *none* represents incompleteness and *both* expresses inconsistency. There is also a sense in which *fuzzy logic* is a paraconsistent logic. For details of paraconsistent logics, the reader is referred to Priest, Routley and Norman (1989). The recent trend of paraconsistent logics may be found in Batens, Mortensen, Priest, and Van Bendegem (2000).

4 Paraconsistent Databases vs. Data Warehouses

We are now ready to give an outline of the basis of data warehouses within the framework of paraconsistent logics. Here, we do not stick to a particular system of paraconsistent logics. And we assume the propositional language (the extension to predicate logic is straightforward). We also employ the logic programming style notation.

First, a *fact* is expressed as a literal of the form A or $\neg A$, namely it is written as:

$$A \leftarrow$$

Second, a *rule* is of the form:

$$A \leftarrow B_1, \ldots, B_n$$

whose interpretation is that if B_1 and...and B_n hold then A holds. This is written in the standard notation as:

$$B_1 \& \ldots \& B_n \rightarrow A$$

Third, an *integrity constraint* is of the form:

$$\leftarrow C_1, \ldots, C_m$$

whose interpretation is that C_1 and ... and C_m do not simultaneously hold. The above three forms constitue the language of logic databases, i.e. Horn clause subset of classical logic. This is exactly the syntax of logic programming language like Prolog, but the underlying logic is classical logic (cf. Gallair and Minker (1978)). Logic database is also called *deductive database*. The advantage of the use of logic programming language is that it acts as both data definition language and data manipulation language with a clear semantics.

Paraconsistent logic database is a logic database whose underlying logic is a paraconsistent logic. Depending on the choice of paraconsistent logics, we may modify the syntax of the language appropriately. But most paraconsistent logics have the logical symbols $\&, \rightarrow, \neg$, and we have no problems. Due to the use of paraconsistent logics, both C and $\neg C$ may hold. In other words, the integrity constraint of the form

$$\leftarrow C, \neg C$$

fails. The integrity constraint is nothing but the law of non-contradiction in the standard logical form. It is well known that there are cases in which the Clark completion gives rises to inconsistency in (classical) logic databases. In such cases, classical logic has no semantics. On the other hand, in paraconsistent logic databases, inconsistency is in general local. Namely, one contradiction is harmless in the sense that other data can be consistently captured. It is also possible to deduce some conclusion from the paraconsistent logic database in such cases.

But, we need to resolve inconsistency in paraconsistent logic. This perhaps needs non-logical mechanism. The paraconsistent logic database thus guarantees the non-valatility in the sense that we need not update data. It also has the integrity, since the integrity constraint in paraconsistent logic plays the role. In addition, we formalize a query language of paraconsistent logic databases. The query in this context is data mining process which can be expressed as logical inferences. Therefore, data mining can be logically expressed. In fact, we know some implementations of paraconsistent logic programming languages like Paralog.

There are several extensions of logic databases. For instance, *disjunctive logic databases* allow disjunction on the head of a rule, i.e. rules are written in non-Horn clauses. By this extension, we can specify some descriptions like null values related to incompleteness in disjunctive logic databases. It would be possible to extend the present framework with disjunction for paraconsistent logic databases.

In this way, if we employ some system of paraconsistent logic as the underlying logic of databases, the database will become a data warehouse. It should be also noted that if inconsistency does not arise in a paraconsisten logic data base, it behaves like the (classical) logic database. In this regard, paraconsistent logic databases are an extension of classical logic databases. But this is not the end of story.

5 Possible Extensions related to KDD

Indeed paraconsistent logic databases can be viewed as data warehouses, but they do not have all the properties given above. To complete our ideas, we need some extensions of the system in the previous section. Data in data warehouse should be subject-oriented. In other words, each data interacts each other depending on some subject matter. For this purpose, relevance logic is of special use. This is because the relevant implication $A \rightarrow B$ consider *some* connection of the antecedent A and the succedent B. Thus, subject-orientedness could be formalized in relevance logic.

To deal with the time-variance, we should expand a paraconsistent logic with temporal features. The resulting logic is *paraconsistent temporal logic* which has some temporal operators. Another way is to furnish a two sorted paraconsistent logic with the sort of time point capable of referring to the time. Such extensions enable us to specify the time of the entry of data helpful to the analysis of data.

If we view data warehouses as a component of decision support system, we must formalize the notion of decision from data warehouses. To achieve the task, the concepts like knowledge and belief are necessary. The requirement is to develop *paraconsistent epistemic (belief) logic.* The so-called BDI architecture can be incorporated in our framework of intensional paraconsistent logics to motivate paraconsistent agent technolog in (distributed) databases. Such *intensional* extensions of paraconsistent logics can strengthen the power of paraconsistent logic databases in relation to data warehouses.

In general, data warehouses cannot update data. But we may be able to update or revise data in paraconsistent logic databases. In the literature on philosophy and AI, *belief revision* has been extensively investigated (cf. Gärdenfors (1988)). In the logic of belief revision, three basic operations called expansion, contraction, and revision are considered to model the dynamics of beliefs. The logic can be also applied to the foundations of data bases. We can reformulate the logic of belief revision in a paraconsistent setting.

6 Conclusions

We proposed a foundation of data warehouses in paraconsistent logics. The attempt may be original and firstly be in print. The basic idea is that data warehouses can be seen as paraconsistent logic databases and the restriction to (classical) logic databases induces several stages related to data mining. This implies that paraconsistent logics can be regarded as one of the sound descriptions of data warehouses with several notions from data mining.

Future work includes the formulation of the present ideas in the context of relational databases. We need to develop a version of paraconsistent relational algebra in the style of Codd (1972). If it is successful, the paraconsistent versions of both domain and tuple logics could be similarly reformulated.

Another topic is to design the language of data warehouses capable of data mining. One of the candidates is obviously *annotated logics* of da Costa et al. (1991). The reason is that annotated logics have smooth basis and computational proof theory.

It would be also interesting to fuse the given foundation with non-logical data mining techniques from statistics, learning, and so on. This line will classify the usefulness of known techniques in logic-based approaches to data warehouses.

Because the present paper addresses the conceptual basis of data warehouses in logical view, we should elaborate on the ideas technically. We hope to report these topics in forthcoming papers.

References

Adriaans, P. and Zantinge, D.: *Data Mining*, Addison-Wesley, Reading, Mass., 1996.

Anderson, R. and Belnap, N.: *Entailment vol. I*, Princeton University Press, Princeton, 1976.

Batens, D., Mortensen, C., Priest, G. and Van Bendegem, J.-P., eds., *Frontiers of Paraconsistent Logic*, Research Studies Press, Baldock, 2000.

Belnap, N.: A useful four-valued logic, J.M. Dunn and G. Epstein (eds.), *Modern Uses of Multiple-Valued Logic*, 8-37, Reidel, Dordrecht, 1977.

Codd, E.: A relational model of data for large shared data banks, *Communications of the ACM 13*, 377-387, 1972.

da Costa, N.: On the theory of inconsistent formal systems, *Notre Dame Journal of Formal Logic 15*, 497-510, 1974.

da Costa, N., Subrahmanian, V. and Vago, C.: The paraconsistent logic $\mathbf{P}\tau$, *Zeitschrift für mathematische Logik und Grundlagen der Mathematik 37*, 139-148, 1991.

da Costa, N., Abe, J. and Subrahmanian, V.: Remarks on annotated logic, *Zeitschrift für mathematische Logik und Grundlagen der Mathematik 37*, 561-570, 1991.
Date, C.: *An Introduction to Database Systems*, Addison-Wesley, Reading, Mass., 2000.
Gallair,H. and Minker, J. (eds.): *Logic and Data Bases*, Plenum Press, New York, 1978.
Gärdenfors, P.: *Knowledge in Flux*, MIT Press, Cambridge, Mass., 1988.
Inmon, W.: *Building Data Warehouse*, Wiley, New York, 1992.
Jaskowski, S.: Propositional calculus for contradictory deductive systems, *Studia Logica 24*, 143-157, 1969 (first published in 1948).
Priest, G., Routley, R. and Norman, J., eds.: *Paraconsistent Logic*, Philosphia Verlag, München, 1989.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Visualization of Class Structures using Piecewise Linear Classifiers

Hiroshi TENMOTO[1] Yasukuni MORI[2] Mineichi KUDO[2]

[1] *Department of Information Engineering*
Kushiro National College of Technology
Otanoshike Nishi 2-32-1, Kushiro 084-0916, Hokkaido, Japan
Tel: +81-154-57-7351 E-mail: `tenmo@kushiro-ct.ac.jp`

[2] *Division of Systems and Information Engineering*
Graduate School of Engineering, Hokkaido University
Kita 13 Nishi 8, Kita-ku, Sapporo 060-8628, Japan
Tel: +81-11-706-6852 E-mail: `{yasu,mine}@main.eng.hokudai.ac.jp`

Abstract. Piecewise linear classifiers are utilized in order to visualize class structures in high-dimensional feature space. The feature space is split into many polyhedral regions by the component hyperplanes of the classifier, and the class label of each polyhedron is determined by taking majority rule on the number of samples. The intra- and inter-class relationships of the polyhedra are shown as a graph, drawing each polyhedron as a node and connecting the nodes by edges according to the adjacency of the regions. Compared to another graph representation method, the proposed method is superior in showing the inseparability or the closeness between the classes.

1 Introduction

In pattern recognition, it is getting to be important to evaluate the sufficiency of the features that are collected in order to classify the observed patterns correctly. However, in general, it is difficult to judge whether the features are enough for the given problem or not, because the dimensionality of the feature space is usually high, so the users cannot recognize the spatial structure of the classes in the feature space. In addition, the class regions are usually complicatedly formed with the other classes, so the simple statistical models, e.g., normal distributions, are insufficient to analyze the class structures.

One of the available way to visualize such high-dimensional class structures is mapping technique. Many mapping techniques have been proposed in order to project high-dimensional supervised data onto two-dimensional space[1, 2, 3]. However, such conventional methods project the individual training samples onto two-dimensional space, so some important discriminant informations, such as the distances or the adjacencies among the samples may be lost. Therefore, the mapped data do not necessarily hold the structural relationships between the classes, so the results are often different from what we need.

On the other hand, Mori, *et al.* proposed an alternative way to visualize the high-dimensional class structures[4]. In the method, they map "subclass" over training samples[5], a kind of overlapping cluster, onto two-dimensional space as a node instead of the individual samples, and connect the nodes by edges that represent the relationships between the subclasses. By this method, they succeeded to visualize the spatial relationship among the class regions without large loss of discriminant information. The comparison between the subclass-based graph-representation method and the conventional mapping techniques was discussed in reference [6].

In this paper, we propose another way to visualize the spatial class structures, in which the clusters of the training samples to be represented as nodes are formed by piecewise linear classifiers. An example of the classifier is shown in Fig.1. In this method, the training samples are split into nonoverlapping clusters by the component hyperplanes. Each cluster consists of samples from almost same class. In this study, we project such clusters in a high-dimensional space onto two-dimensional space in order to visualize the spatial structures of the class regions.

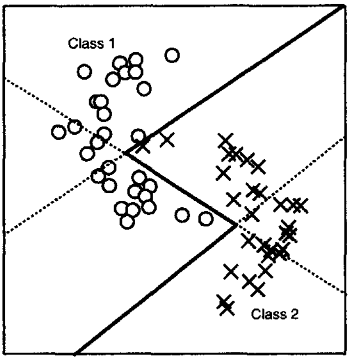


Figure 1: Example of the piecewise linear classifier. The symbols represent training samples and the polygonal solid line represent classification boundary. The dotted lines represent the component hyperplanes that separate the training samples into clusters.

Such a graph representation had been already made by Sklasnky and Michelotti[7]. However, in that paper, the nodes were drawn manually, because the authors' main purpose was simplification of the classifier, not was visualization.

Therefore, in this paper, we try to determine the locations of nodes by the principal two components obtained by KL expansion in order to visualize the graph automatically. In addition, we take the inseparability, or the closeness, of the classes into consideration by representing it as the thickness of the edges.

2 Piecewise Linear Classifier and its Graph Representation

In this study, we employed the Sklansky and Michelotti's method[7] for the construction of piecewise linear classifiers. This method is originally intended to use for two-class problems, but later extended for multiclass problems[8]. The algorithm can be described briefly as follows:

1. Form provisional clusters on the training samples for each class. Any conventional clustering methods can be used for this purpose. We applied $k$means method in this paper.

Keep the resultant assignments of the individual samples to the provisional clusters, and calculate prototypes as the mean vectors of the local samples in the clusters. Let the prototype sets be $\mathcal{M}_1$ and $\mathcal{M}_2$.

Note that, for fine visualization, the number of prototypes for each class should be appropriately selected according to the complexity of the class structure, which is given by a priori information or MDL criterion[9].

2. Find close-opposed pair set Π defined as $\mathcal{L}_{12} \cap \mathcal{L}_{21}$. Here, $\mathcal{L}_{12}$ is the set of prototype-pairs $\{(\boldsymbol{u}, \boldsymbol{v}) | \boldsymbol{u} \in \mathcal{M}_1, \boldsymbol{v} \in \mathcal{M}_2\}$, in which $\boldsymbol{v}$ is the nearest prototype from $\boldsymbol{u}$, and $\mathcal{L}_{21}$ is defined in the reverse way. This concept can be extended to $\Pi^{(k)}$ by expanding the nearness up to the kth. In this study, for the value of k, we adopted the maximum number of prototypes among the two classes.

3. Separate all of the pairs in $\Pi^{(k)}$ by hyperplanes. The hyperplanes have to be placed so as to classify the local training samples correctly. Here, the local samples are constructed by samples that are associated with the prototype-pairs to be cut by this hyperplane.

   Any nonparametric linear classifiers can be employed for this purpose. In the original paper[7], they used a probabilistic descending method called "window training procedure" [10, 11]. However, such a probabilistic algorithm produces different results in each run-time, and such a behavior is not preferable for visualization. Therefore, in this paper, we adopted deterministic "minimum squared error" method[12] for the local training.

4. Determine the class label of each region split by the hyperplanes. Majority rule is applied for the decision on the local training samples in each region.

As a result, piecewise linear decision boundary is formed in the feature space like as Fig.1. Such a boundary may have an appropriate complexity as a classifier.

From the resultant piecewise linear classifier, we can obtain a graph representation $G = (V, E)$ of the high-dimensional class structures as follows:

1. For each region that is surrounded by the component hyperplanes, calculate the mean vector of the local samples. Let the mean vectors as nodes $v \in V$. Here, we calculate the projections of the mean vectors onto two-dimensional space, which is spanned by the two principal basis vectors obtained by KL expansion taken on all of the training samples. The sizes of the nodes are decided proportional to the standard deviation of the local samples in the corresponding regions.

   Here, we introduce a threshold parameter θ in order to suppress the appearance of numerous very small nodes. The nodes that do not have samples more than θ are removed from the resultant graph.

2. Connect the nodes by edges $e \in E$ if the two nodes belong to the same class and the distance between the corresponding regions is 1. Here, the distance is calculated by city block distance, i.e., the adjacent regions sharing same hyperplane have distance 1.

3. Connect the nodes between different classes if the distance between the corresponding regions is 1. Here, the thickness of the edge is determined proportional to the following entropy:

$$\sum_{i=1}^{2} -p_i^+ \log_2 p_i^+ - p_i^- \log_2 p_i^-$$

$$p_i^+ = \frac{N^+}{N^+ + N^-}, \quad p_i^- = \frac{N^-}{N^+ + N^-}.$$

Here, N^+ is the number of training samples belonging to the major class, and N^- is the number of training samples belonging to the minor class (Fig.2).

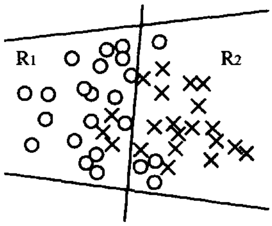


Figure 2: Example of counting samples for entropy calculation. Here, $N^+ = 19, N^- = 3$ for region R_1, and $N^+ = 18, N^- = 4$ for region R_2.

By this conversion, we can obtain the graph-representation of high-dimensional class structures via piecewise linear classifiers. The result for the example data with 100 samples per class is shown in Fig.3. For this dataset, we used five prototypes for each class and 2 for the value of threshold θ. We can recognize that, by this graph, the intra-class structures form two opposite moon-like shapes, and the inter-class closeness is strong around the tips of the two moons.

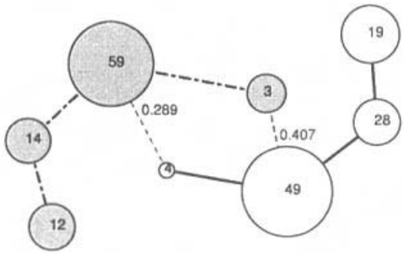


Figure 3: Graph representation for the example data. The white circles represent the clusters of class 1, and the gray circles represent the clusters of class 2. The number of local samples in each cluster is shown in the corresponding node. The inter-class edges with the values of entropies represent the closeness between the classes.

3 Subclass Method and its Graph Representation

In the subclass-based graph-representation method[4], hyperrectangles that surround individual training samples are utilized in order to visualize the class structures. The outline of the method can be described as follows.

The subclass method[5] approximates the class regions by the sets of hyperrectangles on the training samples. Let the training samples of a certain class be S^+, and the samples of the other class be S^-. We call $\boldsymbol{x} \in S^+$ positive samples and $\boldsymbol{y} \in S^-$ negative samples. Then, the subclass method finds a set of hyperrectangles such that each hyperrectangle includes the positive samples maximally while excludes the negative samples. We refer to such a hyperrectangle as a "subclass". An example of the subclass method is shown in Fig.4.

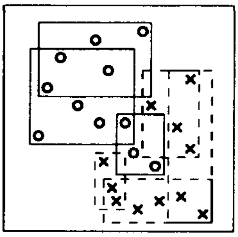


Figure 4: Example of subclass method. The symbols and rectangles represent training samples and subclasses that exclude the negative samples, respectively.

Mori, *et al.* proposed a new method[4] for the visualization of class structures on the basis of this subclass method. They represent the subclasses as a graph $G = (V, E)$ as follows:

1. Represent each subclass s as node $v_s \in V$. The size of $v_s \in V$ is proportional to the volume of s.
2. An edge $e_{s,t} \in E$ between nodes v_s and v_t is drawn only if $o(s,t) > 0$. Here, $o(s,t)$ is the volume of the hyperrectangle intersected by the two subclasses s and t. The width of $e_{s,t}$ is proportional to $o(s,t)$.
3. The following two methods are used in order to determine the locations of nodes $\{v_s\}$.
    - Locate the nodes by the first two principal components in KL expansion of the mean vectors.
    - Locate the nodes on the vertices of a regular polygon.

In order to extract only the important information, noninformative nodes and edges are removed by two thresholds θ_1 and θ_2. Thus, v_s and $e_{s,t}$ are drawn only if

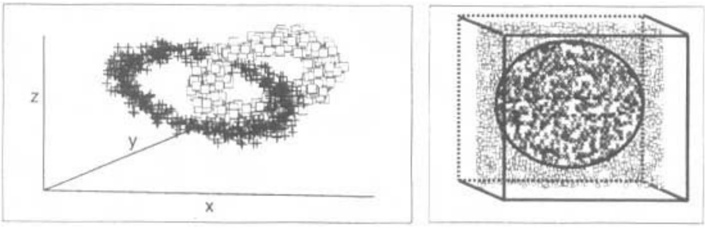
$$\frac{d(s)}{\max_t d(t)} > \theta_1 \quad \text{and} \quad \frac{o(s,t)}{\max_{u,v} o(u,v)} > \theta_2.$$

4 Experimental Comparisons

We carried out experimental comparisons on two cases of artificial datasets. First, we tried to visualize Tori dataset shown in Fig.5(a). This dataset consists of 1000 samples (500 per class) distributing in a three-dimensional artificial feature space. There are two ring structures of different classes. Each class is separated from the other class completely.

The results are shown in Fig.6(a), Fig.6(b) and Fig.6(c). For this dataset, we used ten prototypes for each class and 5 for the value of threshold θ in the proposed method. From the result of the proposed method, we can observe the both class regions form ring shapes, and the two classes encounter each other around the center of the dataset. On the other hand, the two results of the subclass-based methods provide full information of the complete separability of the classes, while the ring structure could not be sufficiently represented.

Next, we had an experiment on a two-class problem in a ten-dimensional artificial feature space, in which the two classes are the inside and outside regions separated by a hypersphere



(a) Tori dataset
(three-dimensional, two-class)

(b) Sphere dataset
(ten-dimensional, two-class)
(drawn in three-dimensional space)

Figure 5: Examples of datasets.

with a volume of 0.5 surrounded by a hypercube with sides of length 1 (Fig5(b)). For this dataset, 1000 data (500 per class) uniformly distributed in both regions were used. Each class is completely separated from the other class also in this dataset.

The results are shown in Fig.6(d), Fig.6(e) and Fig.6(f). For this dataset, we used five prototypes for the inner class, twenty prototypes for the outer class and 13 for the value of threshold θ. We can observe that, from both methods, the inner class is surrounded by the outer class and the inner class is almost unimodal.

For this dataset, we had to use the higher threshold value in order to visualize the inter-class structure clearly. If we take a lower value for the threshold θ, huge number of tiny nodes appear all over the graph and it makes no sense. This is caused by numerous hyperplanes, because we need so many hyperplanes in order to separate such a "complex" dataset in high-dimensional space.

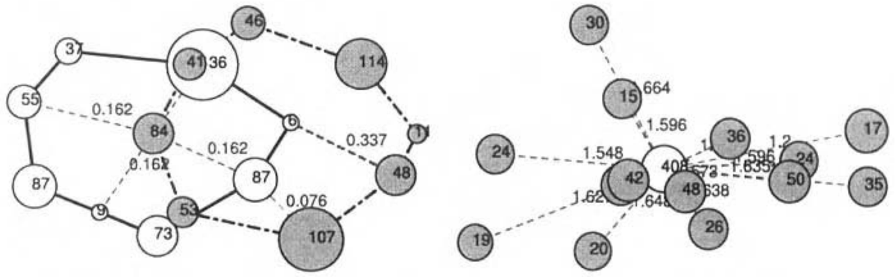
5 Concluding Remarks

We proposed an alternative way to represent the spatial class structures as graphs via piecewise linear classifiers. In this method, the training samples are split into nonoverlapping clusters by the component hyperplanes, and the clusters in a high-dimensional space are projected onto two-dimensional space instead of projecting the individual samples.

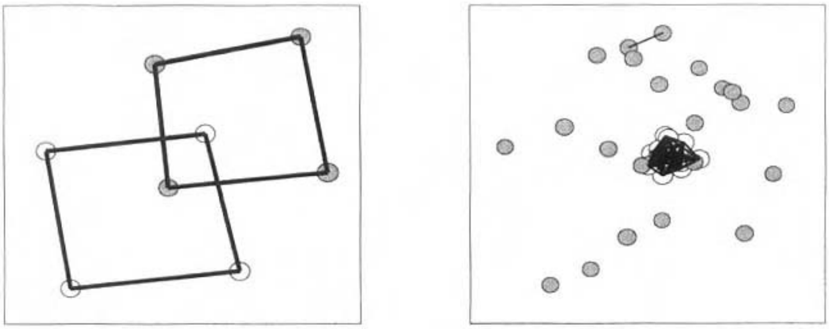
We also tried to determine automatically the locations of vertices by the principal two features obtained by KL expansion. In addition, we took the closeness information between classes into consideration by representing it as the thickness of the edges.

From experimental comparison between the proposed method and the subclass-based graph method, we obtained the following conclusions:

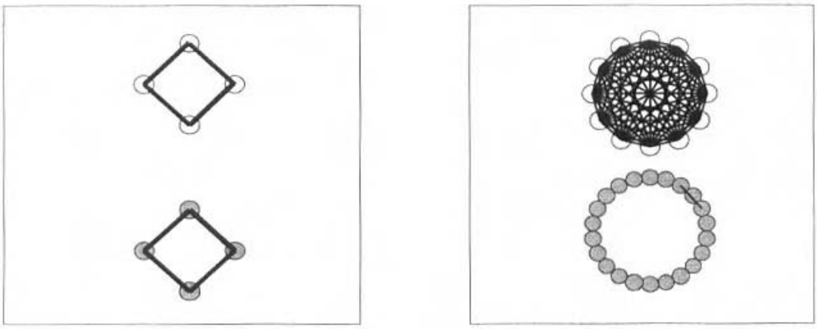
- The proposed method is superior in showing the inseparability or the closeness between classes, because the subclass method tends to avoid overlapping of the different classes in nature. The proposed method produces inter-class edges even for completely separated datasets due to the piecewise linear approximation of class boundaries.
- The proposed method apparently depends on the construction method of piecewise linear classifiers. However, the construction method of the optimal piecewise linear classifier is still being developed. The piecewise linear classifiers with optimal number of hyperplanes[13] should be tested for the visualization.


46
37
41
36
114
55
0.162
84
6
0.337
0.162
0.162
87
48
87
9
53
0.076
73
107
30
15
664
1.596
36
17
24
1.548
1.2
24
42
50
35
48
26
19
20

(a) Proposed Method for Tori dataset (d) Proposed Method for Sphere dataset



(b) Subclass-based Method (KL) for Tori dataset (e) Subclass-based Method (KL) for Sphere dataset



(c) Subclass-based Method (Polygon) for Tori dataset (f) Subclass-based Method (Polygon) for Sphere dataset

Figure 6: Graph representations for Tori dataset and Sphere dataset.

- The proposed method may cause scattering problem, i.e., very small regions may be produced numerously by the component hyperplanes. In order to overcome this difficulty, we need merging such hashed small regions and represent them as virtual large nodes.

There are various visualization methods of high-dimensional supervised data, and each has its own properties. Therefore, the users have to observe and unify the informations obtained by several methods for the given problem.

Acknowledgment

This work was partly supported by Grant-in-Aid for Scientific Research (B), No. 60205101, from Japan Society for the Promotion of Science.

References

[1] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press (1990) 225–257.

[2] M. Aladjem, Parametric and Nonparametric Linear Mappings of Multidimensional Data, *Pattern Recognition* **24** 6(1991) 543–553.

[3] M. Aladjem, Linear Discriminant Analysis for Two Classes via Removal of Classification Structure, *IEEE Transactions of Pattern Analysis and Machine Intelligence* **19** 2(1994) 187–192.

[4] Y. Mori, M. Kudo, J. Toyama and M. Shimbo, Visualization of Classes Using a Graph, *Proceedings of the 14th International Conference on Pattern Recognition (ICPR '98)* **Vol. II**(1998) 1724–1727.

[5] M. Kudo and M. Shimbo, Optimal Subclasses with Dichotomous Variables for Feature Selection and Discriminant, *IEEE Transactions on Systems, Man and Cybernetics* **19**(1989) 1194–1199.

[6] Y. Mori, M. Kudo, J. Toyama and M. Shimbo, Analysis of Pattern Data Using Graph Representation, *Proceedings of the International ICSC Congress on Computational Intelligence: Methods and Applications 2001 (CIMA2001)* (2001) CD-ROM.

[7] J. Sklansky and L. Michelotti, Locally Trained Piecewise Linear Classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2** 2(1980) 101–111.

[8] Y. Park and J. Sklansky, Automated Design of Multiple-Class Piecewise Linear Classifiers, *Journal of Classification* **6**(1989) 195–222.

[9] J. Rissanen, A Universal Prior for Integers and Estimation by Minimum Description Length, *Annals of Statistics* **11**(1983) 416–431.

[10] J. Sklansky and G. N. Wassel, *Pattern Classifiers and Trainable Machines*, Springer-Verlag New York Inc. (1981).

[11] L. Bobrowski and J. Sklansky, Linear Classifiers by Window Training, *IEEE Transactions on Systems, Man and Cybernetics* **25** 1(1995) 1–9.

[12] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification, Second Edition*, John Wiley & Sons, Inc. (2001) 236–238.

[13] H. Tenmoto, M. Kudo and M. Shimbo, Piecewise Linear Classifiers with an Appropriate Number of Hyperplanes, *Pattern Recognition* **31** 11(1998) 1627–1634.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Design of Tree Classifiers Using Interactive Data Exploration

Yasukuni MORI and Mineichi KUDO
Division of Systems and Information Engineering
Graduate School of Engineering
Hokkaido University
Kita-13, Nishi-8, Kita-ku, Sapporo 060-8628, Japan
Tel: +81-11-706-6852 Email:{yasu,mine}@main.eng.hokudai.ac.jp

Abstract. In pattern recognition, knowledge of the structure of pattern data can help us to know the sufficiency of features and to design classifiers. We have proposed a graphical visualization method where the structure and separability of classes in the original feature space are almost correctly preserved. This method is especially effective for knowing which groups of classes are close and which groups are not. In this paper, we propose a method to group classes on the basis of this graphical analysis. Such a grouping is exploited to design a decision tree in which a sample is classified into groups of classes at each node with a different feature subset, and is further divided into smaller groups, and finally reaches at one of the leaves consisting of single classes. The main characteristics of this method is to use different feature subsets in different nodes. This way is most effective to solve multi-class problems. An experiment with 30 characters (30 classes) was conducted to demonstrate the effectiveness of the proposed method.

1 Introduction

Exploratory data analysis is a very important technique in pattern recognition. The main goal of pattern recognition is to determine correctly a class to which a given data belongs, that is, to design a good classifier. A good classifier naturally requires a good feature space. When adequate features are gathered, it is expected that all data belonging to a same class are close to each other and, as a whole, are far from the other classes in the feature space. However, features are usually collected empirically and, thus, it is difficult to collect informative features only. Therefore, it is important to analyze the structure of classes and to investigate the state of scattering of data points in a given feature space. However, since the dimensionality of feature space is generally very high, we cannot perceive the state of these data directly. Mapping techniques onto a low dimensional space are, therefore, introduced in order to help our understanding. A comparative study by the authors showed that a graphical visualization of data subsets is most effective to capture faithfully the structure of data of a class and the separability of classes [1, 2]. In addition, this method allows us to see which groups of classes are close to each other and which groups are not. Therefore, we can easily group classes (clustering of classes).

On the other hand, *feature selection* is known to be very effective to improve the performance of classifiers designed from a finite number of samples. So far many studies have been devoted to develop the methodology (for example, [3, 4]). Removing useless features for classification, we can raise up the estimation precision of parameters of parametric classifiers and can avoid abuse of no-informative features even in non-parametric classifiers. As a result, it is expected that the generalization error is decreased by feature selection. Usually a feature subset is chosen in common to all classes. However, in multi-class problems, it is natural to think that the most informative feature subsets are different depending on classes to be classified. For example, in a pair of similar characters such that a short stroke at a certain location exists in one character and does not exist in the other character, the existence of the stroke is the key for distinguishing these two characters, but such a key is useless for distinguishing a different pair of characters. So, the authors have been discussing how to choose different feature subsets in different groups of classes [5]. Choosing different feature subsets depending on different groups of classes naturally results in a decision tree classifier, in which at *root* node the set of all classes are divided into several groups of classes by such a feature subset by which this rough classification is carried out most effectively, and in each *children* nodes, the classes are further divided into finer groups with a feature subset different from the feature subset used in the root node. This is repeated until division produces single classes.

In such a decision tree with different feature subsets, the main difficulty is in how to determine the structure of the tree. For example, at root, when we want to divide C classes into two groups, there are $2^{C-1} - 1$ possible candidates of groupings. This is infeasible even for not so large C. Therefore, in [5], we proposed a bottom-up way to construct a decision tree. However, this approach does not necessarily work well. Thus, we consider to group classes on the basis of the above-mentioned graphical analysis [1]. In this case, a user can help grouping by observing the graphical results. In what follows, we discuss the effectiveness of this approach.

2 Graphical Visualization

First, we describe how we can group classes in order to construct a decision tree. Once the structure of the tree is determined, we carry out feature selection in each node of the tree.

The graph visualization method [1] is based on the subclass method [6, 7] which approximates a true class region by a set of convex hulls spanned by subsets of training samples. In this approach, we find quasi convex hulls, called *subclasses*, including only training samples of a class maximally and excluding the samples of the other classes. An example is shown in Fig. 1. We can know the state of the distributions of the data points from information on these convex hulls, for example, from the volumes of convex hulls and the volume of the intersection between two convex hulls. Using this kind of information, we have proposed a method for visualizing high-dimensional data [2]. Such subclasses can be thought of as a compressed expression of the training samples. For visualization, the following information is extracted from the constructed quasi convex hulls.

1. $p(s)$: the center vector of a subclass s.
2. $V(s)$: the volume of s measured on a normalized discrete domain $[0,1]^D$, where D is calculated by $D = \sum_{i=1}^{k} 2^i \binom{d}{i}$ for d original features and k (usually $=$ 2 or 3) is a resolution parameter (for details, see [7]). The volume $V(s)$ is calculated by

$$V(s) = \left(\prod_{i=1}^{D} |M_i(s) - m_i(s)| \right)^{\frac{1}{D}} \in [0,1],$$

where $m_i(s)$ and $M_i(s)$ are the minimum and the maximum ends, respectively, of s in the ith dimension of D dimensions.

3. $o(s,t)$: the ratio of the volume of the convex region intersected by two subclasses, s and t, to the volume of the union of s and t, that is, $$o(s,t) = \frac{V(s \cap t)}{V(s \cup t)}.$$

4. $n(s)$: the ratio of the number of samples included in s to the total number of samples of the class.

5. $d(s)$: the density of s, that is, $d(s) \doteq n(s)/V(s)$.

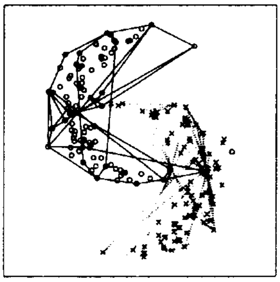


Figure 1: Example of subclasses.

¡ Next, a graph $G = (V, E)$ is made as follows. One vertex $v_s \in V$ corresponds to a subclass s, and one link $\epsilon_{s,t} \in E$ between v_s and v_t is drawn only if $o(s,t) > 0$. Two display types have been proposed. First display type is called a *principal component display* (PCD). In this type, each vertices are located by the coordinates of the first two principle components in Karhunen-Loéve expansion of the center vectors $\{p(s)\}$ (Fig. 2). Second display type is called a *polygon display* (PD). In this type, they are located on the vertices of a regular polygon subclass by subclass and class by class (Fig. 3). The radius r_s of vertex v_s is taken in proportion to $V(s)$. Also, the width w_{st} of link $\epsilon_{s,t}$ is taken to be proportional to $o(s,t)$. PCD would help us to understand the relationship of positions of classes in the original space. While, PD would help us for knowing the degree of overlaps of the class distributions. In order to represent only important information, non-informative vertices or links, or both, are removed by two thresholds, θ_v and θ_l. Thus, a vertex v_s and a link $\epsilon_{s,t}$ are drawn in graph G if and only if

$$\frac{d(s)}{\max_u d(u)} > \theta_v \quad \text{and} \quad \frac{o(s,t)}{\max_{u,v} o(u,v)} > \theta_l.$$

By varying the values of these two thresholds, we can examine the data interactively.

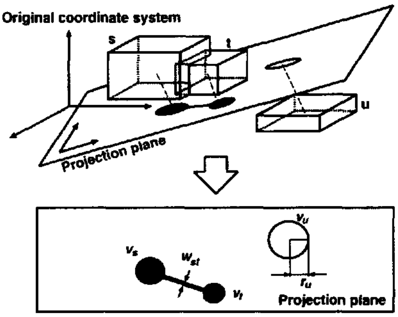


Figure 2: Graph representation in PCD when there are two subclasses s and t for class 1 and subclass u for class 2.

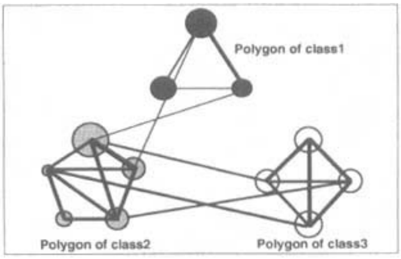


Figure 3: Graph representation in PD when there are 3, 5 and 4 subclasses for class 1, 2 and 3, respectively.

3 Construction of Trees

Now, we can determine the structure of a decision tree in the following manner.

Step 1: With $\theta_l = 1.0$ (no link appears), decreasing gradually the value of θ_v step by step with 0.01 from $\theta_v = 1.0$, find the maximum value of θ_v such that at least one vertex (subclass) of every class appears in the graph.

Step 2: Keeping the value of θ_v as it was, set the value of θ_l to zero. At this stage, some classes are connected to the other classes by some links as long as the subclasses of those classes overlap. Then, increase the value of θ_l gradually until a few groups, hopefully two or three, of classes appear due to vanishment of links that corresponds to a little amount of overlap. At this stage, we can find the first grouping corresponding to the root of the tree.

Step 3: Let us consider each group (a child node of the root node) found in Step 2. Then, increasing the value of θ_l further, we find a finer grouping in which some classes are further separated. This process is applied for all groups (all children nodes of the root).

Step 4: In each of newly created children nodes, Step 3 is repeated until all classes are separated. At this stage, we reach at one of leaves consisting of single classes.

Once a tree is constructed, the next task is to find a best feature subset in each node of the tree. For this purpose, we used an approach based on the structural indices of categories [8]. This method works in a linear order of the number of features and find a feature subset that does not depend a specific classifier. So, for large-problems in size of feature set, this method is appropriate. For the details, see [8].

4 Character Recognition

We performed an experiment in order to evaluate the proposed method. The Japanese character dataset taken from ETL9B database [9] was used. The number of the samples is 200 per class, and we divided them into halves for training and testing. The number of features is 196.

The dataset is of 30 hand-written Japanese characters that are 15 pairs of *similar characters* (Table 1). Here, the word *similar characters* is used for a pair of characters that are similar in shape as one can see, not for showing a pair that are *close* in the feature space. The features were measured as follows. First, we divide a character image of size 64x64 into 49 blocks of size 16x16 in which adjacent blocks overlap a half. Next, in four directions of multiples of 45 degree, we count the number of pixels along to each direction. As a result, $196 (= 4 \times 49)$ features are measured [10]. According to our previous experiments [2], this data was analyzed in some stages with different parameter sets of (θ_l, θ_v) (Fig. 4). From Fig. 4, we can see

Table 1: 15 pairs of similar characters.

「え」 and 「之」	「ば」 and 「ぱ」	「び」 and 「ぴ」
「穎」 and 「頴」	「干」 and 「千」	「熊」 and 「態」
「采」 and 「釆」	「士」 and 「土」	「師」 and 「帥」
「情」 and 「精」	「伸」 and 「仲」	「推」 and 「椎」
「磨」 and 「麿」	「末」 and 「未」	「肋」 and 「助」

1. All classes are represented by single vertices and, thus, are almost separable (Fig. 4(a-1),(a-2)).

2. Similar character pairs tend to be misclassified (Fig. 4(b-1),(b-2)).

3. Simple characters and complex ones form two groups (Fig. 4(c-2)).

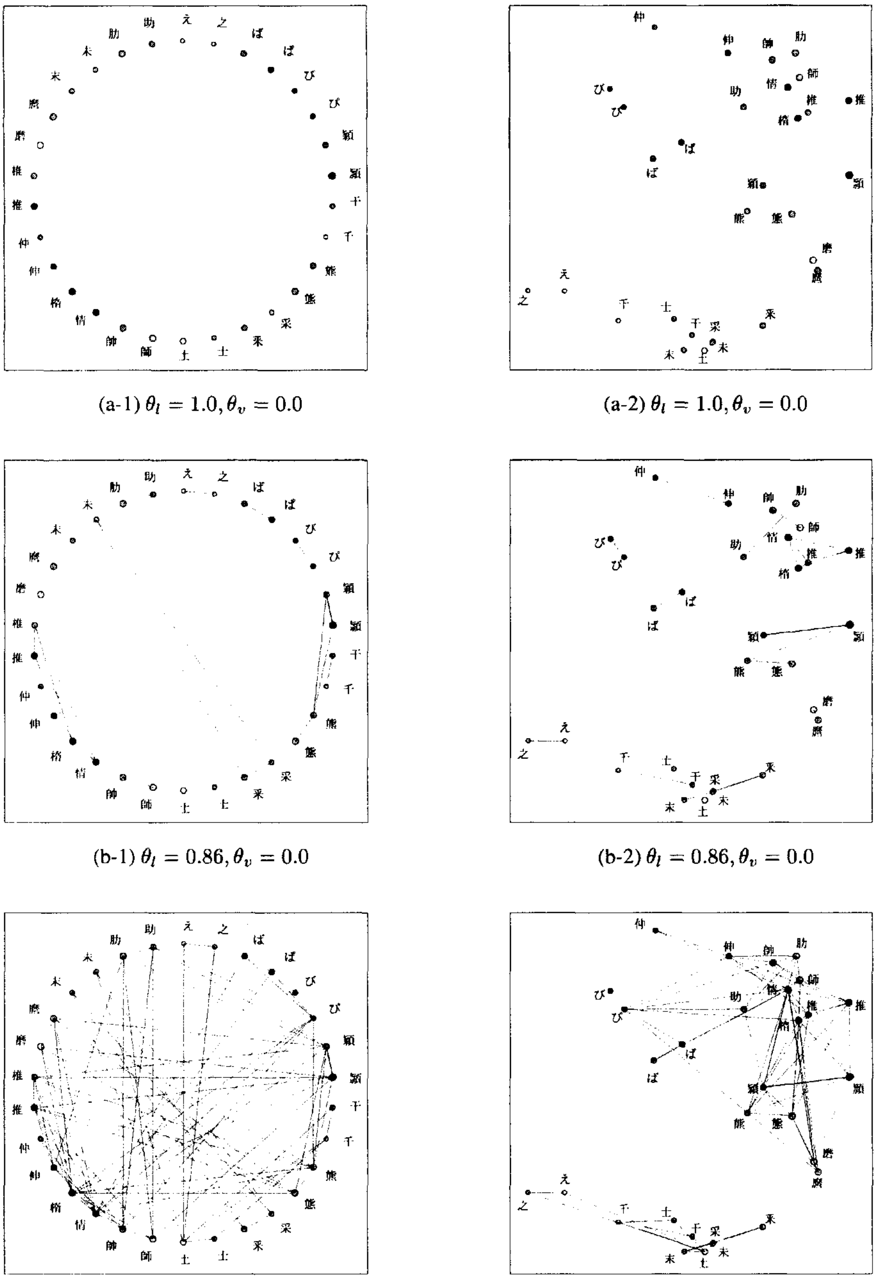
For grouping of all classes in a hierarchical way, we have to start with $\theta_l = 0.0$ (Fig. 4(c)) and to raise the value of θ_l gradually. These steps are partly visualized in Fig. 5. The final decision tree is shown in Fig. 6. In an inner node, the number in the circle shows the number of features used in the node, and in a leaf the class character is shown. The size of feature subsets was reduced to 130.8 from 196 on the average.

We used the nearest neighbor (1-NN) classifier and the plug-in Bayes linear classifier classifier as the element classifiers in all nodes in the decision tree. For the linear classifier, we replaced the eigen values smaller than 100.0 of the estimated covariance matrix with 100.0 for avoiding over-estimation.

The recognition rates of the 1-NN classifier and the linear classifier were 80.5 % and 90.3 %, respectively, when all 196 features were used. With the decision tree, they were improved to 81.4 % and 91.9 %, respectively. In nodes A, B and C in Fig. 6, a large reduction is found in the number of features. The selected features are visualized in Fig. 7. In Fig. 7, a selected feature is connected to a short line segment presented at the corresponding location with the corresponding direction. For example, when feature-no. 6 is selected, it is displayed as a vertical line segment in the second block of the top row. This figure makes it clear that in node A, the upper-left part of images is most important for classification, in node B, slanting lines in the central part, and, in node C, middle-right part, respectively. In fact, the local recognition rate at node C was improved from 92.0 % to 96.0 % in the linear classifier.

5 Conclusion

We proposed a method for constructing a decision tree. The structure is determined by a user in support of graphical visualization analysis of data points. The main characteristic of the



(a-1) $\theta_l = 1.0, \theta_v = 0.0$

(a-2) $\theta_l = 1.0, \theta_v = 0.0$

(b-1) $\theta_l = 0.86, \theta_v = 0.0$

(b-2) $\theta_l = 0.86, \theta_v = 0.0$

(c-1) $\theta_l = 0.00, \theta_v = 0.0$

(c-2) $\theta_l = 0.00, \theta_v = 0.0$

Figure 4: Results for Japanese character dataset. There are 15 pairs of similar characters. Left figures are PD's and right figures are PCD's. For readability, all links are drawn with a same width.

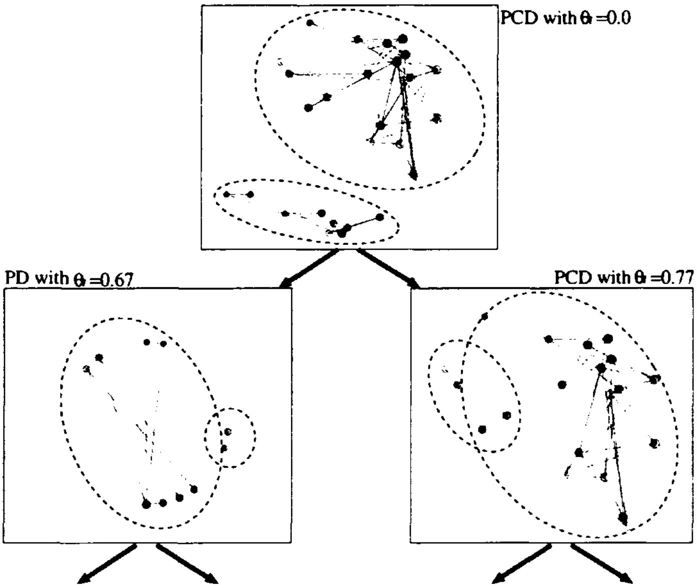


Figure 5: Grouping at top level and at 2nd level.

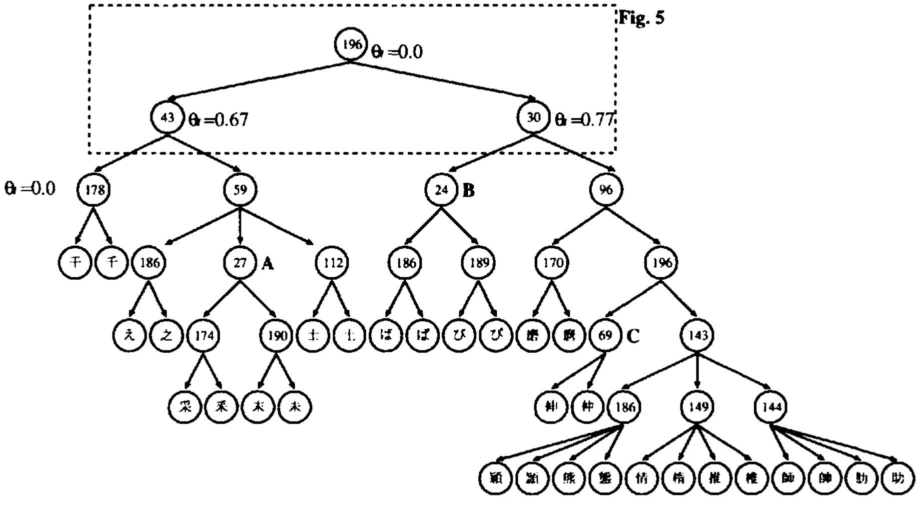
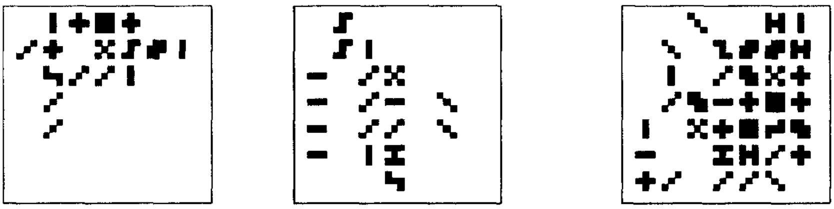


Fig. 5
196 θ=0.0
43 θ=0.67
30 θ=0.77
θ=0.0
178
59
24 B
96
186
27 A
112
186
189
170
196
え
之
174
190
69 C
143
186
149
144

Figure 6: Decision tree of Japanese character dataset. The number in each node shows the number of features.



(a) in node A (b) in node B (c) in node C

Figure 7: Feature subsets chosen in three nodes in Fig. 6.

tree is that in each node a different feature subset is chosen so as to maximize the performance of the local classification at that node. The effectiveness of this approach was demonstrated in a Japanese character recognition problem.

Acknowledgment

This work was partly supported by Grant-in-Aid for Scientific Research (B), No 60205101, Japan.

References

[1] Y. Mori, M. Kudo, J. Toyama and M. Shimbo, Comparison of Low-Dimensional Mapping Techniques Based on Discriminatory Information, Proceeding of 2nd International ICSC Symposium on Advances in Intelligent Data Analysis (AIDA'2001), United Kingdom(2001) CD-ROM Paper-No 1724-166.

[2] Y. Mori and M. Kudo, Interactive Data Exploration Using Graph Representation, Proceeding of 6th World Multiconference on Systems, Cybernetics and Informatics (SCI2002), Florida(2002), to appear.

[3] J. Kittler, Feature set search algorithms. in C. H. Chen (ed.), Pattern Recognition and Signal Processing, Sijthoff and Noordhoff, Alphen aan den Rijn, Netherlands (1978) 41–60.

[4] P. Pudil, J. Novovičová, and J. Kittler, Floating Search Methods in Feature Selection, Pattern Recognition Letters, **15** (1994) 1119–1125.

[5] K. Aoki and M. Kudo, Decision Tree Using Class-Dependent Feature Selection, Proceeding of 4th International Workshop on Statistical Techniques in Pattern Recognition (SPR2002), Canada(2002), to appear.

[6] M. Kudo, S. Yanagi and M. Shimbo, Construction of Class Regions by a Randomized Algorithm: A Randomized Subclass Method, Pattern Recognition, **29** (1996) 581-588.

[7] M. Kudo, Y. Torii, Y. Mori, and M.Shimbo, Approximation of Class Regions by Quasi Convex Hulls, Pattern Recognition Letters, **19** (1998) 777-786.

[8] M. Kudo and M. Shimbo, Feature Selection Based on the Structural Indices of Categories, Pattern Recognition, **26** (1993) 891–901.

[9] T. Saito, H. Yamada and K. Yamamoto, On the Data Base ETL9 of Handprinted Characters in JIS Chinese Characters and Its Analysis, The Transactions of the Institute of Electronics and Communication Engineers of Japan, **J68-D** (1985) 757–764 (In Japanese).

[10] N. Sun, M. Abe and Y. Nemoto, A Handwritten Character Recognition System by Using Improved Directional Element Feature and Subspace Method, The Transactions of the Institute of Electronics, Information and Communication Engineers, **J78-D-II** (1995) 922–930 (In Japanese).

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Clustering Based on Gap and Structure

Mineichi KUDO
Division of Systems and Information Engineering
Graduate School of Engineering
Hokkaido University, Sapporo 060-8628, JAPAN
mine@main.eng.hokudai.ac.jp

Abstract. Proposed is a clustering algorithm that takes into consideration the gap among data points and the structure of data points. In a previous study, we observed that two different clusters are separated by a gap and that a geometrical structure is important to form a cluster. Based on this observation, we 1) made the gap appear by noise points placed around data points and 2) extracted a geometrical structure as a classification boundary in a problem of distinguishing the data class from the noise class. However, this method is applicable only to two-dimensional data. In this study, we present a method that eliminates this limitation. Some experimental results show the effectiveness of this approach.

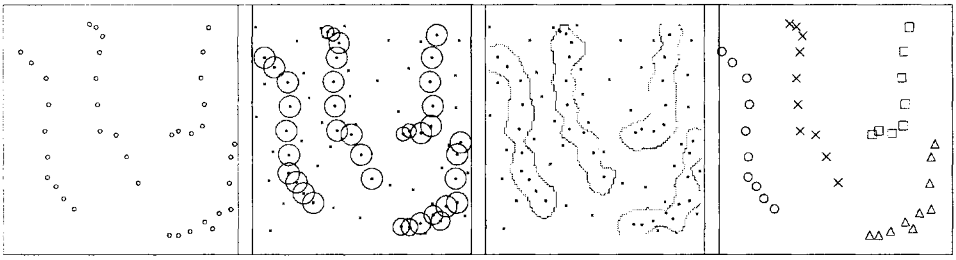
1 Introduction

Many clustering algorithms have been proposed [1, 2, 3, 4]. One group of algorithms, including k-means [1] and fuzzy c-means [2] algorithms, is suitable for finding spherical clusters, while another group, such as the algorithms proposed in Refs. [3, 4], is suitable for handling arched clusters. However, it is difficult to construct algorithms that meet both requirements.

Any algorithm produces a single result as long as its parameters are fixed. However, according to human clustering results in 2-dimensional data, the clustering results seem not to be unique. This is often observed in the case of small dataset sizes. A possible reason for this is that the way of feeling gaps between data points differs depending on the observer or that the structure found in data points differs depending on the observer. In our previous study, we considered these two factors and proposed an algorithm that reflects them [5]. The proposed algorithm has the following merits: 1) it produces several clustering results as human observers, 2) the number of clusters is determined automatically, and 3) it can explain the change in results according to a change in the size of the background space on which points are presented. However, the algorithm is limited to two-dimensional data. In this paper, we present an extension of the algorithm that enables higher-dimensional data to be handled.

2 Algorithm

First, we show an outline of the algorithm with an example (Fig. 1). The algorithm consists of the following three stages.



(a) Original data

(b) Avoidance fields and noise points

(c) Classification boundary

(d) Clustering result (# of clusters is 4)

Figure 1: Outline of the algorithm.

Stage 1 Generate some noise points, avoiding near the data points (Fig. 1(b)).

Stage 2 Solve a two-class problem in which one class is of data points and the other class is of noise points (Fig.1 (c)).

Stage 3 Find the connected regions in the class region of data points (Fig. 1(d)).

In our previous study [5], such an *avoidance field* for generation of noise points was given a physiological interpretation. There is a field called a *foveal visual field* that is the range that we can see without eye movement when we focus our eyes on a single point on a sheet, and the size of this field is considered to be 1°20′, corresponding to the size of a *fovea*. It should be noted that this is an angle, and the absolute range (radius) is determined by the distance between the point and the observer. In other words, a foveal visual field around a point can be seen the area occupied by that point. In addition, we assume that such a foveal visual field is affected by the density around the point. That is, the visual foveal fields becomes smaller as the density increases (Fig. 2). This idea is based on a perception model in a 2-dimensional case. However, the distance between a point and an observer can be replaced by the relationship between the size of the data area and the size of the background area, e.g., the ratio of the diameter of the smallest sphere containing a whole data points to the diameter of the background space (a sheet in 2-dimensional cases). In this way, we can apply the idea of the foveal visual fields to a case of more than two dimensions.

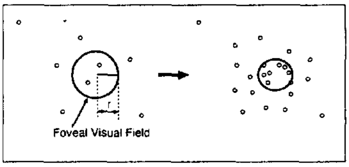


Figure 2: The density effect of a foveral visual field.

In Stage 1, an avoidance field (circle) around a data point is determined by two factors: the size of the background area and the density of the local area around the point. We generate noise points in such a way that 1) they are generated randomly according to a uniform distribution, 2) do not fall into the avoidance fields, and 3) keep a certain distance from each other. The radius r of the avoidance field around a point x is determined by

$$r = \min(r_p, r_d),$$

where $r_p = \alpha \times \beta \times d_s$ and $r_d = d_{1-NN}(x)$. Here, $d_{1-NN}(x)$ is the distance of x to its nearest neighbor and d_s is the length of the diagonal of the minimum covering square of a whole data points. The value of r_p is calculated by multiplying a margin rate β for determining the size of the background space, and a foveral ratio α. The value of r_p corresponds to the size of the foveal visual field, and the value of r_d reflects the density effect.

In Stage 2, a classifier is used to solve the two-class problem. Such a classifier has to be flexible enough for expressing multi-modal distributions.

Stage 3 has so far been solved as a labeling problem using an image processing technique.

3 Extension to a higher dimension

The previously proposed method was limited to two-dimensional data, because the stage for extracting clusters (Stage 3) could not be processed in a higher dimensionality. After Stage 2, since a data class region is known, class assignment is easily done for any point. However, finding clusters as connected regions in the data class region requires a special technique.

To cope with this difficulty, we take the following approach.

1. Select any pair of data points and see if the line segment connecting these pair of points goes across the noise class region or not. If it does not, the pair of points is considered to belong to the same connected region, and they are said to be in a relation R.
2. Clusters are found by taking a transitive closure of R.

In practice, the line segment is replaced by a set of points on that line. To check if two points are in R or not, we take the following approach. For two points x and y, the middle point $z_1 = (x+y)/2$ is first checked and then two points $z_2 = (x+z_1)/2$ and $z_3 = (z_1+y)/2$ are checked, and so on. This procedure is repeated recursively up to depth d, that is, $2^d - 1$ equally-spaced points on the line segment are examined. This recursive call is terminated when a point belonging to the noise class is found (Fig. 3). In this way, it is possible that a connected region is judged as more than one region. Fig. 3 is such an example. This is a limitation of this procedure. The precision, however, will increase by increasing the number of data points and the value of d.

4 Experiments

We used the support vector machine (SVM) [7] with radial basis functions as its kernel functions. The SVM has two parameters: c for specifying the degree of soft margin and σ for specifying the degree of sharpness in the kernels. We adopted $c = 1000$, which means that a hard margin is expected, because, in our setting, the classification problem is separable in nature. The value of σ was chosen so as to give an appropriate result in each experiment. The parameters for avoidance fields α and β were set to $\alpha = 0.035$ and $\beta = 1.1$, respectively. We generated the same number of noise points as the number of data points.

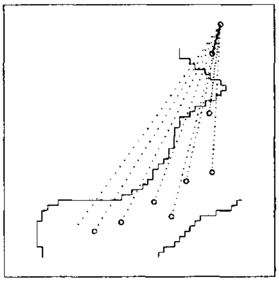
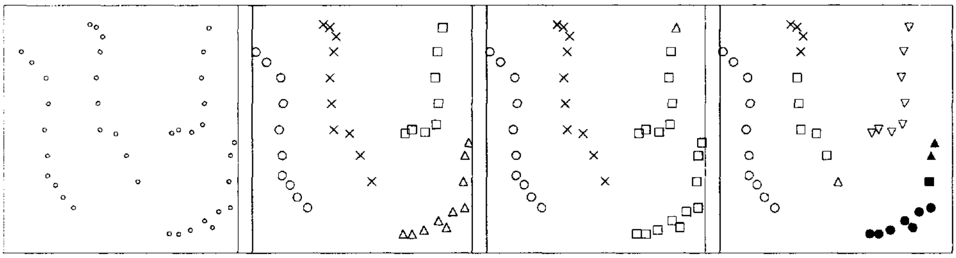


Figure 3: Finding clusters.

We dealt with three kinds of synthetic datasets: two in a two-dimensional and one in a three-dimensional spaces.

(1) Synthetic dataset 1

We used a dataset similar to those in [4]. Many of human observers may find four line-shaped clusters in this dataset. The results are shown in Fig. 4. The desired result (Fig. 4(b)) was obtained four times in 20 trials. Various results (4 to 7 clusters) were obtained. This variation in the results is caused by different sets of noise points. All of the results seem reasonable in different interpretations.



(a) Original data (b) Result 1 (3/20 trials) (c) Result 2 (2/20 trials) (d) Result 3 (2/20 trials)

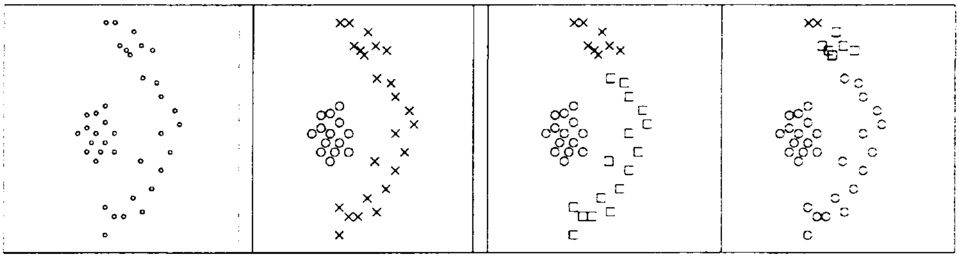
Figure 4: Results for dataset 1 ($\sigma = 5.0$). Different symbols show different clusters.

(2) Synthetic dataset 2

This dataset (Fig. 5(a)) seems to be more difficult than dataset 1 to cluster. Many human observers may find two clusters in this data. Three typical results with their frequencies of occurrence in 20 trials are shown in Fig. 5.

(3) Synthetic Data 3 (Tori data)

This dataset consists of two tori coupled orthogonally in untouched way (for details, see [6]) (Fig. 6(a)). The dataset consists of 200 samples (100 per torus). The results are shown in Fig. 6. In our approach, all 10 trials produced the same desirable result. The k-means and fuzzy c-means with a power parameter $m = 1.6, 2.0$ were also examined (Fig. 6(c)). All of those results were almost the same.



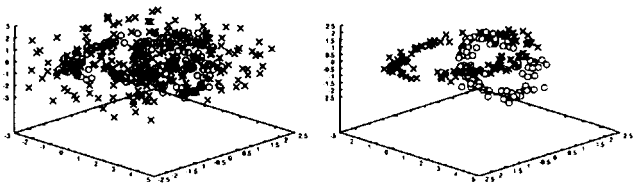
(a) Original data

(b) Result 1 (4/20 trials)

(c) Result 2 (3/20 trials)

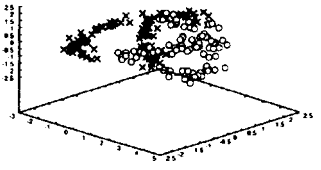
(d) Result 3 (1/20 trials)

Figure 5: Results for dataset 2 ($\sigma = 0.5$). Different symbols show different clusters.



(a) Original data and noise

(b) Result by proposed method (10/10 trials)



(c) Result by Fuzzy c-means method ($m = 2.0$)

Figure 6: Results for dataset 3 ($\sigma = 0.5$). Different symbols show different clusters.

5 Discussion

The proposed algorithm differs from previous algorithms in the following points: 1) the algorithm takes into consideration the gap and the structure of data points, 2) clustering is formulated as classification that distinguishes data points from noise points, and 3) the algorithm produces several different results even in a fixed parameter set. Property (3) is the most remarkable characteristic of this approach. In the case of a small dataset, the various results give several different but meaningful interpretations of the dataset. In the case of a large dataset, there would be less variation in the results. In addition, if every clusters is aggregate and a large gap exists among clusters, the results have less variation, say, a unique result. In that case, the results would not be sensitive to the parameters.

A change in the parameters of the classifier results in a different cluster number. In the case of SVM, a larger value of σ produces many clusters. If this algorithm is carried out with a sequence of decreasing values of σ, it gives a hierarchical clustering.

6 Conclusion

We have proposed a clustering algorithm that is based on gap and structure of data points. This algorithm is an extension of our previously proposed algorithm. Experimental results using the newly proposed algorithm were satisfactory in the sense that the most frequently obtained results were consist with the clustering results by human observers or were natural for the datasets. The following problems remain to be solved: 1) clarification of the relationship between the number of noise points and the clustering results and 2) improvement in a systematic way to choose the values of parameters of a classifier.

References

[1] M. R. Anderberg. *Cluster Analysis for Applications*. Academic Press, 1973.
[2] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum, New York, 1981.
[3] A. Fred. Finding consistent clusters in data partitions. In *Multiple Classifier Systems*, Lecture Notes in Computer Science, pages 309–318. Springer, 2001.
[4] L. Y. Tseng and S. B. Yang. A genetic clustering algorithm for data with non-spherical-shape clusters. *Pattern Recognition*, 33:1251–1259, 2000.
[5] M. Kudo, T. Murai, and M. Shimbo. Clustering consistent with human perception. Proceedings of the *Second International ICSC Symposium on Advances in Intelligent Data Analysis*, (CD-ROM) paper-no 1724–168, 2001.
[6] J. Sklansky and L. Michelotti. Locally trained piecewise linear classifiers. *IEEE Trans. Pattern Analy. Machine Intelli.*, PAMI-2:101–111, 1980.
[7] V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Tables in Relational Databases from a Point of View of Possible-Worlds-Restriction

Tetsuya MURAI[1], Michinori NAKATA[2], and Yoshiharu SATO[2]

1) *Division of Systems and Information Engineering, Graduate School of Engineering, Hokkaido University, Kita 13, Nishi 8, Kita-ku, Sapporo 060-8628, JAPAN, E-mail: {murahiko, ysato}@main.eng.hokudai.ac.jp*

2) *Faculty of Management and Information Sciences, Josai International University, 1 Gumyo, Togane, Chiba 283-8555, JAPAN, E-mail: nakata@jiu.ac.jp*

Abstract. Data in databases are assumed to be represented as atomic sentences in a logical language. Then, basic tables are generated by a given data as a set of atomic sentences. Further compound tables are also constructed by combining atomic sentences with conjunction. A kind of uncertainty, incompleteness of a given data, is explicated when compound tables are constructed, thus lower and upper approximations in rough set theory is introduced to deal with the incompleteness. Then null values are formally represented as ones in the borderline region of approximation.

1 Introduction

Relational databases[3] and their logic-based extension known as deductive databases are a powerful approach to intelligent database and knowledge-based systems. For the authors, however, their main defect seems to be that they start from tables, that is, there are first of all tables, then logical concepts are introduced for deduction. Then their constructs seem to be too complicated typically like so many kinds of normalization. In this paper, we present a first step for a logic-based approach to databases using possible-worlds-restriction[4]. In the approach, there are first of all atomic sentences as data and then tables are introduced only for the purpose of representing data.

2 Possible-Worlds-Restriction

Let $\mathfrak{P} = \{\mathsf{p}_1, \cdots, \mathsf{p}_n\}$ be a finite set of atomic sentences A modal language $\mathfrak{L}(\mathfrak{P})$ is formed from $\mathfrak{P}$ in the usual way of combining logical operators such as $\neg$ (negation), and (conjunction), or (disjunction), $\rightarrow$ (material implication), $\leftrightarrow$ (equivalence), and **B** (belief). Then we can generate 2^n possible worlds

$$w_k : \pi_1^k \wedge \cdots \wedge \pi_n^k \quad (1 \leq k \leq 2^n),$$

where π_i^k is exactly one of p_i and $\neg\mathsf{p}_i$. They are nothing but state-descriptions defined by Carnap[1]. Let $W_{\mathfrak{P}}$ be a set of all such worlds generated from $\mathfrak{P}$. We can identify the set of possible worlds with the power set of $\mathfrak{P}$ by a bijection $w_k \rightarrow \{\mathsf{p}_i \mid \pi_i^k = \mathsf{p}_i\}$ in the following way: $W_{\mathfrak{P}} = 2^{\mathfrak{P}}$. In crisp cases, one of those 2^n worlds should be exactly the actual one. If, however, we do not have so much information about p_i, then we cannot specify which of the worlds is the actual one.

Example 1 *Let* $\mathfrak{P} = \{\mathsf{p}_1, \mathsf{p}_2, \mathsf{p}_3\}$ *, then we have the following* $2^3(=8)$ *possible worlds and their corresponding sets:*

Possible Worlds	Logical Sentences	Sets
w_1	$\mathsf{p}_1\wedge \mathsf{p}_2\wedge \mathsf{p}_3$	$\{\mathsf{p}_1,\mathsf{p}_2,\mathsf{p}_3\}$
w_2	$\mathsf{p}_1\wedge \mathsf{p}_2\wedge\neg\mathsf{p}_3$	$\{\mathsf{p}_1,\mathsf{p}_2\}$
w_3	$\mathsf{p}_1\wedge\neg\mathsf{p}_2\wedge \mathsf{p}_3$	$\{\mathsf{p}_1,\mathsf{p}_3\}$
w_4	$\mathsf{p}_1\wedge\neg\mathsf{p}_2\wedge\neg\mathsf{p}_3$	$\{\mathsf{p}_1\}$
w_5	$\neg\mathsf{p}_1\wedge \mathsf{p}_2\wedge \mathsf{p}_3$	$\{\mathsf{p}_2,\mathsf{p}_3\}$
w_6	$\neg\mathsf{p}_1\wedge \mathsf{p}_2\wedge\neg\mathsf{p}_3$	$\{\mathsf{p}_2\}$
w_7	$\neg\mathsf{p}_1\wedge\neg\mathsf{p}_2\wedge \mathsf{p}_3$	$\{\mathsf{p}_3\}$
w_8	$\neg\mathsf{p}_1\wedge\neg\mathsf{p}_2\wedge\neg\mathsf{p}_3$	$\emptyset$

∎

Definition 2 Given a set of atomic sentences $\mathfrak{P}$, a *possible-worlds-restriction model* (pwr-model, for short) for $\mathfrak{P}$ is defined as the following three tuple:

$$\mathfrak{M}_{\mathfrak{P}} = <W_{\mathfrak{P}}, B_{\mathfrak{P}}, \models>,$$

where $W_{\mathfrak{P}} = 2^{\mathfrak{P}}$, $B_{\mathfrak{P}}(\neq \emptyset) \subseteq W_{\mathfrak{P}}$, and $\mathfrak{M}_{\mathfrak{P}}, w \models \mathsf{p}$ (for $\mathsf{p} \in \mathfrak{P}$) iff $\mathsf{p} \in w$. ∎

In the definition, $B_{\mathfrak{P}}$ is a non-empty set of subsets of $W_{\mathfrak{P}}$ whose elements are possible candidates for the actual world under available information. The relation $\models$ can be extended for any sentences in $\mathfrak{L}(\mathfrak{P})$ in a usual way. In particular, we can define a truth-condition for the modal operator by

$$\mathfrak{M}_{\mathfrak{P}}, w \models \mathbf{B}\mathsf{p} \overset{\text{def}}{\Leftrightarrow} \forall w' (w' \in B_{\mathfrak{P}} \Rightarrow \mathfrak{M}_{\mathfrak{P}}, w' \models \mathsf{p}).$$

In general, the proposition $||\mathsf{p}||^{\mathfrak{M}}$ of a sentence p in a model $\mathfrak{M}$ is defined as the set of possible worlds where p is true:

$$||\mathsf{p}||^{\mathfrak{M}} \overset{\text{def}}{=} \{w \mid \mathfrak{M}, w \models \mathsf{p}\}\ .$$

Then it is easy to see the above truth condition can be written

$$\mathfrak{M}_{\mathfrak{P}}, w \models \mathbf{B}\mathsf{p} \Longleftrightarrow B_{\mathfrak{P}} \subseteq ||\mathsf{p}||^{\mathfrak{M}_{\mathfrak{P}}}.$$

In the model, we can define an *accessibility relation* R between possible worlds by

$$wRw' \Longleftrightarrow w' \in B_{\mathfrak{P}}.$$

Thus pwr-models are a standard kind of Kripke models, at least $KD45$-models, where

K. $\mathbf{B}(\mathsf{p}\rightarrow\mathsf{p}')\rightarrow(\mathbf{B}\mathsf{p}\rightarrow\mathbf{B}\mathsf{p}')$, **D.** $\mathbf{B}\mathsf{p}\rightarrow\neg\mathbf{B}\neg\mathsf{p}$,
4. $\mathbf{B}\mathsf{p}\rightarrow\mathbf{B}\mathbf{B}\mathsf{p}$, **5.** $\neg\mathbf{B}\neg\mathsf{p}\rightarrow\mathbf{B}\neg\mathbf{B}\neg\mathsf{p}$,

because the relation R satisfies seriality, transitivity, and euclidness (cf.[2]).

If $\mathfrak{M}_{\mathfrak{P}}, w \models \mathsf{p}$ holds at any world w in a model $\mathfrak{M}_{\mathfrak{P}}$, p is said to be *valid* in this model and denoted $\mathfrak{M}_{\mathfrak{P}} \models \mathsf{p}$.

Example 3 *For the set of possible worlds described in Example 1, assume that we have two pieces of information I_1: p_1 and I_2: $\mathsf{p}_1 \rightarrow \mathsf{p}_3$. Then they are translated into the following restrictions:*

$$\begin{aligned}\mathfrak{R}_1 &= \|\mathsf{p}_1\| = \{w_1, w_2, w_3, w_4\} ,\\ \mathfrak{R}_2 &= \|\mathsf{p}_1 \rightarrow \mathsf{p}_3\| = \{w_1, w_3, w_5, w_6, w_7, w_8\} ,\end{aligned}$$

respectively. Then, by taking the intersection of the two pieces of information, we have

$$B_{\mathfrak{P}} = \bigcap_{k=1,2} \mathfrak{R}_k = \{w_1, w_3\} .$$

That is, these two worlds can be accessible.

3 Databases as a Set of Sentences

Let *PRED* be a set of *predicate* symbols. We assume each predicate symbol $\mathsf{p} \in \mathit{PRED}$ has their own number $n(\mathsf{p})$ of *arity*, by which we classify predicate symbols : let

$$\mathit{PRED}_k \stackrel{\text{def}}{=} \{\mathsf{p} \mid n(\mathsf{p}) = k\} ,$$

then

$$\begin{aligned}&k \neq k' \Longrightarrow \mathit{PRED}_k \cap \mathit{PRED}_{k'} = \emptyset,\\ &\mathit{PRED} = \textstyle\bigcup_{k \in \mathbf{N}^+} \mathit{PRED}_k,\end{aligned}$$

where $\mathbf{N}^+$ is the set of positive integers. We denote a predicate symbol p of arity k also by $\mathsf{p}(x_1, \cdots, x_k)$ using variables $x_1, \cdots .x_k$.

Further we assume each place in each predicate symbol has its own set, whose elements can be applicable :

$$\exists U_1^{\mathsf{p}}, \cdots, \exists U_k^{\mathsf{p}} \subseteq U \text{ s.t. } \mathsf{p}(x_1, \cdots, x_k) \Longleftrightarrow x_1 \in U_1^{\mathsf{p}}, \cdots, x_k \in U_k^{\mathsf{p}}.$$

Let $U = \bigcup_{k \in N^+} \bigcup_{\mathsf{p} \in \mathit{PRED}_k} (U_1^{\mathsf{p}} \cup \cdots \cup U_k^{\mathsf{p}})$.

Definition 4 A tuple $<\mathit{PRED}, U>$ is called a (*database*) *frame*. ∎

So, given a frame $\mathfrak{F}=<\mathit{PRED}, U>$, the set $\mathfrak{P}_{\mathfrak{F}}$ of atomic sentences is defined by

$$\mathfrak{P}_{\mathfrak{F}} \stackrel{\text{def}}{=} \bigcup_{k \in N^+} \bigcup_{\mathsf{p} \in \mathit{PRED}_k} \mathfrak{P}_{\mathsf{p}},$$

where

$$\mathfrak{P}_{\mathsf{p}} \stackrel{\text{def}}{=} \{\mathsf{p}(x_1, \cdots, x_k) \mid x_1 \in U_1^{\mathsf{p}}, \cdots, x_k \in U_k^{\mathsf{p}}\}$$

for $\mathsf{p} \in \mathit{PRED}_k$.

Example 5 *Let PRED = PRED₁ ∪ PRED₂, where* $PRED_1 = \{\mathbf{company}, \mathbf{city}, \mathbf{part}\}$, *and* $PRED_2 = \{\mathbf{locate}, \mathbf{supply}\}$. *Also let*

$$\begin{aligned}
&U_1^{\mathbf{company}} = U_1^{\mathbf{locate}} = U_1^{\mathbf{supply}} = U_1 = \{\mathsf{Smith}, \mathsf{Jones}, \mathsf{Adams}\},\\
&U_1^{\mathbf{city}} = U_2^{\mathbf{locate}} = U_2 = \{\mathsf{London}, \mathsf{Paris}\},\\
&U_1^{\mathbf{part}} = U_2^{\mathbf{supply}} = U_3 = \{\mathsf{bolt}, \mathsf{nut}\},\\
&U = U_1 \cup U_2 \cup U_3.
\end{aligned}$$

Thus we have a frame $\mathfrak{F} = <PRED, U>$. *Then we have the following set of atomic sentences:*

$$\begin{aligned}
\mathfrak{P} = \{ \;& \mathbf{company}(\mathsf{Smith}), \mathbf{company}(\mathsf{Jones}), \mathbf{company}(\mathsf{Adams}),\\
& \mathbf{city}(\mathsf{London}), \mathbf{city}(\mathsf{Paris}), \mathbf{part}(\mathsf{bolt}), \mathbf{part}(\mathsf{nut}),\\
& \mathbf{locate}(\mathsf{Smith}, \mathsf{London}), \mathbf{locate}(\mathsf{Smith}, \mathsf{Paris}), \mathbf{locate}(\mathsf{Jones}, \mathsf{London}),\\
& \mathbf{locate}(\mathsf{Jones}, \mathsf{Paris}), \mathbf{locate}(\mathsf{Adams}, \mathsf{London}), \mathbf{locate}(\mathsf{Adams}, \mathsf{Paris}),\\
& \mathbf{supply}(\mathsf{Smith}, \mathsf{bolt}), \mathbf{supply}(\mathsf{Smith}, \mathsf{nut}), \mathbf{supply}(\mathsf{Jones}, \mathsf{bolt}),\\
& \mathbf{supply}(\mathsf{Jones}, \mathsf{nut}), \mathbf{supply}(\mathsf{Adams}, \mathsf{bolt}), \mathbf{supply}(\mathsf{Adams}, \mathsf{nut}) \;\}. \;\blacksquare
\end{aligned}$$

Definition 6 Given a frame $\mathfrak{F}$, any subset $DB \subseteq \mathfrak{P}_{\mathfrak{F}}$ is called a *database in* $\mathfrak{F}$. ■

Example 7 *The following set of atomic sentences is a database in* $\mathfrak{F}$ *:*

$$\begin{aligned}
DB = \{ \;& \mathbf{company}(\mathsf{Smith}), \mathbf{company}(\mathsf{Jones}), \mathbf{locate}(\mathsf{Smith}, \mathsf{London}),\\
& \mathbf{locate}(\mathsf{Jones}, \mathsf{Paris}), \mathbf{supply}(\mathsf{Jones}, \mathsf{nut}) \;\}. \;\blacksquare
\end{aligned}$$

Let $PRED_{DB}$ be a set of predicate symbols that appears in *DB*.

Definition 8 Any element in $PRED_{DB}$ is called a *basic schema in DB*. ■

Example 9 *We have the following set of basic schemata :*

$$PRED_{DB} = \{\mathbf{company}, \mathbf{locate}, \mathbf{supply}\} . \;\blacksquare$$

Definition 10 For $\mathsf{p} \in PRED_{DB}$,

$$DB_{\mathsf{p}} \overset{\text{def}}{=} DB \cap \mathfrak{P}_{\mathsf{p}},$$

is called a *subdatabase of DB with respect to* p. ■

Obviously we have $DB = \biguplus_{\mathsf{p} \in PRED_{DB}} DB_{\mathsf{p}}$, where $\biguplus$ is the directsum.

4 Basic Tables

First we construct a pwr-model for a basic schema p :

$$\mathfrak{M}_{\mathsf{p}} = < W_{\mathsf{p}}, B_{\mathsf{p}}, \models >,$$

where $W_{\mathsf{p}} = 2^{\mathfrak{P}_{\mathsf{p}}}$, $B_{\mathsf{p}} = \{w \in 2^{\mathfrak{P}_{\mathsf{p}}} \mid DB_{\mathsf{p}} \subseteq w\}$, and $\mathfrak{M}_{\mathsf{p}}, w \models \mathsf{p}$ (for $\mathsf{p} \in \mathfrak{P}_{\mathsf{p}}$) iff $\mathsf{p} \in w$. Given a basic schema p and its model $\mathfrak{M}_{\mathsf{p}}$, we can define a *table* T_{p} generated in $\mathfrak{M}_{\mathsf{p}}$ as a set of data that have support by belief.

Definition 11 Given a frame $\mathfrak{F}$ and a database *DB* in $\mathfrak{F}$, a *basic table with respect to* p *generated in* $\mathfrak{M}_{\mathsf{p}}$ is define by

$$\mathsf{T}_{\mathsf{p}} \overset{\text{def}}{=} \{(x_1, \cdots, x_n) \mid \mathfrak{M}_{\mathsf{p}} \models \mathbf{B}\mathsf{p}(x_1, \cdots, x_n)\} \ . \ \blacksquare$$

By the property of the pwr-model as *KD*45-models, the following lemma holds :

Lemma 12 *Let* $t = (x_1, \cdots, x_n)$*, then*

$$\begin{aligned} t \notin \mathsf{T}_{\mathsf{p}} \quad &\Leftrightarrow \quad \mathfrak{M}_{\mathsf{p}} \models \neg\mathbf{B}\mathsf{p}(t) \\ &\Leftrightarrow \quad \mathfrak{M}_{\mathsf{p}} \models \neg\mathbf{B}\neg\mathsf{p}(t) \wedge \neg\mathbf{B}\mathsf{p}(t)) \text{ or } \mathfrak{M}_{\mathsf{p}} \models \mathbf{B}\neg\mathsf{p}(t). \ \blacksquare \end{aligned}$$

Now we can formally understand that tuples which do not appear in a table have the status of either *unknown* ($\neg\mathbf{B}\neg\mathsf{p} \wedge \neg\mathbf{B}\mathsf{p}$) or *negative belief* ($\mathbf{B}\neg\mathsf{p}$). In what follows we use the following abbreviation

$$\mathbf{U}\mathsf{p} \overset{\text{def}}{\longleftrightarrow} \neg\mathbf{B}\neg\mathsf{p} \wedge \neg\mathbf{B}\mathsf{p},$$

which corresponds to *contingency* in the usual modal logic. We show examples of tables generated in pwr-models.

Example 13 *Let* $\mathfrak{P}_{\mathsf{company}} = \{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3\}$*, where* $\mathsf{c}_1 = \mathsf{company(Smith)}$, $\mathsf{c}_2 = \mathsf{company(Jones)}$, *and* $\mathsf{c}_3 = \mathsf{company(Adams)}$. *Then they generate the following set of worlds* $W_{\mathsf{company}} = \{w_1, \cdots, w_8\}$, *where* $w_1 = \{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3\}$, $w_2 = \{\mathsf{c}_1, \mathsf{c}_2\}$, $w_3 = \{\mathsf{c}_1, \mathsf{c}_3\}$, $w_4 = \{\mathsf{c}_1\}$, $w_5 = \{\mathsf{c}_2, \mathsf{c}_3\}$, $w_6 = \{\mathsf{c}_2\}$, $w_7 = \{\mathsf{c}_3\}$, *and* $w_8 = \emptyset$. *The subdatabase* $DB_{\mathsf{company}} = \{\mathsf{c}_1, \mathsf{c}_2\}$ *restricts the set of possible worlds to* $\mathcal{B}_{\mathsf{company}} = \{w_1, w_2\}$, *because*

$$DB_{\mathsf{company}} \subseteq w_1 = \{\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3\} \ , \quad DB_{\mathsf{company}} \subseteq w_2 = \{\mathsf{c}_1, \mathsf{c}_2\}$$

so we have the following valid belief sentences :

$$\begin{aligned} &\mathfrak{M}_{\mathsf{company}} \models \mathbf{B}\mathsf{company(Smith)}, \\ &\mathfrak{M}_{\mathsf{company}} \models \mathbf{B}\mathsf{company(Jones)}, \\ &\mathfrak{M}_{\mathsf{company}} \models \mathbf{U}\mathsf{company(Adams)}. \end{aligned}$$

Thus, by Definition 11, we have the following basic tables.

$\mathsf{T}_{\mathsf{company}}$

company
Smith
Jones

In the similar way, from two subdatabases

$$\begin{aligned} DB_{\mathsf{locate}} &= \{\mathsf{locate(Smith, London)}, \mathsf{l}_4 = \mathsf{locate(Jones, Paris)}\} \\ DB_{\mathsf{supply}} &= \{\mathsf{supply(Smith, bolt)}\} \end{aligned}$$

we have the following other two basic tables.

$\mathsf{T}_{\mathsf{locate}}$

company	city
Smith	London
Jones	Paris

$\mathsf{T}_{\mathsf{supply}}$

company	part
Smith	bolt

Thus our database

$$\begin{aligned} DB = \{ \ &\mathsf{company(Smith)}, \mathsf{company(Jones)}, \mathsf{locate(Smith, London)}, \\ &\mathsf{locate(Jones, Paris)}, \mathsf{supply(Smith, bolt)} \ \} \end{aligned}$$

is represented by the following set of three tables: $\{\mathsf{T}_{\mathsf{locate}}, \mathsf{T}_{\mathsf{company}}, \mathsf{T}_{\mathsf{supply}}\}$.

5 Compound Tables

In general, given a database DB, we define a set of schemata SCH_{DB} in DB is defined in the following recursive way:

$$\begin{array}{ll} (1) & \mathsf{p}(x_1, \cdots, x_k) \in PRED_{DB} \Rightarrow \mathsf{p}(x_1, \cdots, x_k) \in SCH_{DB}, \\ (2) & \mathsf{p}(\alpha_1, \cdots, \alpha_k), \mathsf{p}'(\beta_1, \cdots, \beta_{k'}) \in SCH_{DB} \text{ and } (U_i^{\mathsf{p}} = U_j^{\mathsf{p}'} \text{ for some } i, j) \\ & \Rightarrow (\mathsf{p} \wedge \mathsf{p}')(\gamma_1, \cdots, \gamma_{k''}) \in SCH_{DB}, \end{array}$$

where α_i, β_j, and γ_k are meta-variables for variables $x_1, x_2, \cdots$ and

$$\begin{array}{l} U_i^{\mathsf{p}} = U_j^{\mathsf{p}'} \Rightarrow \alpha_i = \beta_j, \\ \{\gamma_1, \cdots, \gamma_{k''}\} = \{\alpha_1, \cdots, \alpha_k\} \cup \{\beta_1, \cdots, \beta_{k'}\}. \end{array}$$

A schema is called *compound* when it is not a basic one.

Example 14 *We can make the following compound schemata:*

$$\begin{array}{l} (\mathbf{company} \wedge \mathbf{locate})(x_1, x_2), \\ (\mathbf{company} \wedge \mathbf{supply})(x_1, x_3), \\ (\mathbf{locate} \wedge \mathbf{supply})(x_1, x_2, x_3). \quad \blacksquare \end{array}$$

Let us consider generation of tables from compound schemata.

Again we want to construct a pwr-model $\mathfrak{M}_{\mathsf{p} \wedge \mathsf{p}'}$ for a compound schema $\mathsf{p} \wedge \mathsf{p}'$, where assume we make

$$(\mathsf{p} \wedge \mathsf{p}')(x''_1, \cdots, x''_{k''})$$

from $\mathsf{p}(x_1, \cdots, x_k)$ and $\mathsf{p}'(x'_1, \cdots, x'_{k'})$. We often identify $\mathsf{p} \wedge \mathsf{p}'$ with set $\{\mathsf{p}, \mathsf{p}'\}$.

The set of possible worlds and a valuation is defined in a similar way:

$$\begin{array}{l} W_{\mathsf{p} \wedge \mathsf{p}'} = 2^{\mathfrak{P}_{\mathsf{p} \wedge \mathsf{p}'}}, \\ \mathfrak{M}_{\mathsf{p} \wedge \mathsf{p}'}, w \models \mathsf{p} \ \ (\text{for } \mathsf{p} \in \mathfrak{P}_{\mathsf{p} \wedge \mathsf{p}'}) \Leftrightarrow \mathsf{p} \in w. \end{array}$$

To define the set of accessible worlds $B_{\mathsf{p} \wedge \mathsf{p}'}$, we must notice that a database is given as a set of atomic sentences

$$DB_{\mathsf{p},\mathsf{p}'} = DB_{\mathsf{p}} \cup DB_{\mathsf{p}'},$$

while each world is a set of compound sentences that has form of the compound schema in question. So we must embed the database $DB_{\mathsf{p},\mathsf{p}'}$ into the set of compound sentences $\mathfrak{P}\mathsf{p} \wedge \mathsf{p}'$. To accomplish this, we use approximation in rough set theory[5], in the following two way :

$$\begin{array}{rcl} \overline{\mathcal{X}}_{\mathsf{p} \wedge \mathsf{p}'} & \stackrel{\text{def}}{=} & \{X \subseteq \mathfrak{P}_{\mathsf{p} \wedge \mathsf{p}'} \mid \cup X \subseteq DB_{\mathsf{p},\mathsf{p}'}\}, \\ \overline{DB}_{\mathsf{p} \wedge \mathsf{p}'} & \stackrel{\text{def}}{=} & \{X \in \overline{\mathcal{X}} \mid \exists Y \in \overline{\mathcal{X}}(Y \subseteq X) \Rightarrow X = Y\}, \\ \underline{\mathcal{X}}_{\mathsf{p} \wedge \mathsf{p}'} & \stackrel{\text{def}}{=} & \{X \subseteq \mathfrak{P}_{\mathsf{p} \wedge \mathsf{p}'} \mid DB_{\mathsf{p},\mathsf{p}'} \subseteq \cup X\}, \\ \underline{DB}_{\mathsf{p} \wedge \mathsf{p}'} & \stackrel{\text{def}}{=} & \{X \in \underline{\mathcal{X}} \mid \exists Y \in \underline{\mathcal{X}}(X \subseteq Y) \Rightarrow X = Y\}. \end{array}$$

Then we can define the following two kinds of sets of accessible worlds :

$$\overline{B}_{\mathsf{p}\wedge\mathsf{p}'} = \{\ \{w \in 2^{\mathfrak{P}_{\mathsf{p}\wedge\mathsf{p}'}} \mid X \subseteq w\}\ \}_{X \in \overline{DB}_{\mathsf{p}\wedge\mathsf{p}'}},$$
$$\underline{B}_{\mathsf{p}\wedge\mathsf{p}'} = \{\ \{w \in 2^{\mathfrak{P}_{\mathsf{p}\wedge\mathsf{p}'}} \mid X \subseteq w\}\ \}_{X \in \underline{DB}_{\mathsf{p}\wedge\mathsf{p}'}}.$$

Thus we have a *multi-pwr-model*[1] for a compound schema $\mathsf{p}\wedge\mathsf{p}'$:

$$\mathfrak{M}_{\mathsf{p}\wedge\mathsf{p}'} = <W_{\mathsf{p}\wedge\mathsf{p}'},\ \{\overline{B}_{\mathsf{p}\wedge\mathsf{p}'}, \underline{B}_{\mathsf{p}\wedge\mathsf{p}'}\}\ ,\ \models>.$$

Then we can define two kinds of belief operators:

$$\mathfrak{M}_{\mathsf{p}\wedge\mathsf{p}'}, w \models \mathbf{B}\mathsf{q} \overset{\text{def}}{\Longleftrightarrow} \exists W' \in \underline{B}_{\mathsf{p}\wedge\mathsf{p}'}(W' \subseteq \|\mathsf{q}\|),$$
$$\mathfrak{M}_{\mathsf{p}\wedge\mathsf{p}'}, w \models \mathrm{B}\mathsf{q} \overset{\text{def}}{\Longleftrightarrow} \exists W' \in \overline{B}_{\mathsf{p}\wedge\mathsf{p}'}(W' \subseteq \|\mathsf{q}\|),$$

and we have

$$\mathbf{B}\mathsf{q} \to \mathrm{B}\mathsf{q} \to \neg\mathrm{B}\neg\mathsf{q} \to \neg\mathbf{B}\neg\mathsf{q}.$$

Thus we can define two kinds of compound tables :

Definition 15 A *compound table with respect to* $\mathsf{p}\wedge\mathsf{p}'$ *generated in* $\mathfrak{M}_{\mathsf{p}\wedge\mathsf{p}'}$ is define by

$$\underline{\mathsf{T}}_{\mathsf{p}\wedge\mathsf{p}'} \overset{\text{def}}{=} \{(x_1, \cdots, x_n) \mid \mathfrak{M}_{\mathsf{p}} \models \mathbf{B}\mathsf{p}\wedge\mathsf{p}'(x_1, \cdots, x_n)\}\ ,$$
$$\overline{\mathsf{T}}_{\mathsf{p}\wedge\mathsf{p}'} \overset{\text{def}}{=} \{(x_1, \cdots, x_n) \mid \mathfrak{M}_{\mathsf{p}} \models \mathrm{B}\mathsf{p}\wedge\mathsf{p}'(x_1, \cdots, x_n)\}\ .\ \blacksquare$$

Obviously we have

$$\underline{\mathsf{T}}_{\mathsf{p}\wedge\mathsf{p}'} \subseteq \overline{\mathsf{T}}_{\mathsf{p}\wedge\mathsf{p}'}.$$

So the elements in set-difference $\overline{\mathsf{T}}_{\mathsf{p}\wedge\mathsf{p}'} \setminus \underline{\mathsf{T}}_{\mathsf{p}\wedge\mathsf{p}'}$ can be candidates for unknown data we may specify among many other unknown ones.

Example 16 *Let* $\mathfrak{P}_{\textbf{company}\wedge\textbf{supply}} = \{\mathsf{CS}_{11}, \mathsf{CS}_{12}, \mathsf{CS}_{23}, \mathsf{CS}_{24}, \mathsf{CS}_{35}, \mathsf{CS}_{36}\}$, *where* CS_{11} = (company∧supply)(Smith, bolt), CS_{12} = (company∧supply)(Smith, nut), CS_{23} = (company∧supply)(Jones, bolt), CS_{24} = (company∧supply)(Jones, nut), CS_{35} = (company∧supply)(Adams, bolt), *and* CS_{36} = (company∧supply)(Adams, nut). *They similarly generate* $2^6 (= 64)$ *set of worlds. The subdatabase is*

$$DB_{\textbf{company,supply}} = \{\textsf{company(Smith)}, \textsf{company(Jones)}, \textsf{supply(Smith, bolt)}\}\ .$$

from which we have

$$\overline{DB}_{\textbf{company}\wedge\textbf{supply}} = \{\ \{\mathsf{CS}_{11}, \mathsf{CS}_{23}\}\ ,\ \{\mathsf{CS}_{11}, \mathsf{CS}_{24}\}\ \},$$
$$\underline{DB}_{\textbf{company}\wedge\textbf{supply}} = \{\ \{\mathsf{CS}_{11}\}\ \}$$

---

[1] This is a multi-Scott-Montague model.

Thus we have the following valid belief sentences :

$$\begin{aligned}
\mathfrak{M}_{\mathbf{company}\wedge\mathbf{supply}} &\models \mathrm{B}(\mathbf{company}\wedge\mathbf{supply})(\mathsf{Smith},\mathsf{bolt}),\\
\mathfrak{M}_{\mathbf{company}\wedge\mathbf{supply}} &\models \mathrm{U}(\mathbf{company}\wedge\mathbf{supply})(\mathsf{Smith},\mathsf{nut}),\\
\mathfrak{M}_{\mathbf{company}\wedge\mathbf{supply}} &\models \mathit{B}(\mathbf{company}\wedge\mathbf{supply})(\mathsf{Jones},\mathsf{bolt}),\\
\mathfrak{M}_{\mathbf{company}\wedge\mathbf{supply}} &\models \mathit{B}(\mathbf{company}\wedge\mathbf{supply})(\mathsf{Jones},\mathsf{nut}),\\
\mathfrak{M}_{\mathbf{company}\wedge\mathbf{supply}} &\models \mathrm{U}(\mathbf{company}\wedge\mathbf{supply})(\mathsf{Adams},\mathsf{bolt}),\\
\mathfrak{M}_{\mathbf{company}\wedge\mathbf{supply}} &\models \mathrm{U}(\mathbf{company}\wedge\mathbf{supply})(\mathsf{Adams},\mathsf{nut}).
\end{aligned}$$

Thus we have the following two compound tables.

$\underline{\mathsf{T}}_{\mathbf{company}\wedge\mathbf{supply}}$

company	**city**	*Status*
Smith	bolt	**B**

$\overline{\mathsf{T}}_{\mathbf{company}\wedge\mathbf{supply}}$

company	**part**	*Status*
Smith	bolt	**B**
Jones	bolt	*B*
Jones	nut	*B*

In the first table $\underline{\mathsf{T}}_{\mathbf{company}\wedge\mathbf{supply}}$*, there is loss of information, i.e.,* **company**(Jones). *In the second table* $\overline{\mathsf{T}}_{\mathbf{company}\wedge\mathbf{supply}}$*, where in the latter table we add a new row of 'Status' for convenience, we can aggregate the second and third tuples if we define the value '-' (null, unknown) if*

$$\{x \mid \mathfrak{M}_{\mathbf{company}\wedge\mathbf{supply}} \models \mathit{B}(\mathbf{company}\wedge\mathbf{supply})(\mathsf{Jones},x)\} = U_3.$$

Then we have the following new table representation

$\tilde{\mathsf{T}}_{\mathbf{company}\wedge\mathbf{supply}}$

company	**part**
Smith	bolt
Jones	–

Thus the information that is lost in the first table is recovered. ∎

6 Concluding Remarks and Acknowledgments

In this paper, we illustrated a way of constructing tables from data as atomic sentences with approximation in rough set theory. The first author was partially supported by Grant-in-Aid No. 14380171 for Scientific Research(B) of the Japan Society for the Promotion of Science of Japan.

References

[1] R.Carnap, *Meaning and Necessity : A Study in Semantics and Modal Logic*. Univ. of Chicago Press, 1947.

[2] B.F.Chellas, *Modal Logic : An Introduction*. Cambridge Univ. Press, 1980.

[3] E.F.Codd, A Relational Model of Data for Large Shared Data Banks. CACM, 13(1970), 377-387.

[4] T.Murai and M.Shimbo, Fuzzy Sets and Modal Logic Based on Evidence. Proc. 6th IFSA Congress, Saõ Paulo, 1995, 471-474.

[5] Z.Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer, 1991.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

On Some Different Interpretations of the Generalized Modus Ponens using Type-2 Fuzzy Sets

Helmut Thiele
University of Dortmund, Department of Computer Science I,
D-44227 Dortmund, Germany
Tel.: +49 231 755 6152
Fax: +49 231 755 6555
E-Mail: **thiele@ls1.cs.uni-dortmund.de**
WWW: **http://ls1-www.cs.uni-dortmund.de**

Abstract. Firstly, we repeat the definition of the general concept of a C^T-Type-2 Fuzzy Set on U introduced in [17].

Secondly, we recall the Generalized Modus Ponens formulated in [17] for C^T-Type-2 Fuzzy Sets on U where this special Generalized Modus Ponens is shortly called C^T-Type-2 GMP. After this we define a General C^T-Type-2 Semantics and develop a Semantic Interpretation of the C^T-Type-2 GMP using a General C^T-Type-2 Semantics. Two so-called Correctness Theorems are proved.

In the third section we discuss operations with C^T-Type-2 Fuzzy Sets on U and reduce these operations to operations with the values of such fuzzy sets where these values are called C^T-Type-2 Fuzzy Values. We introduce two kinds of operations with such values, the so-called external and internal operations following the approach introduced in [16] for operating with context dependent fuzzy sets.

In the fourth section we present four new kinds of interpreting the C^T-Type-2 GMP based on external and internal operations with C^T-Type-2 Fuzzy Values.

The last session (Conclusions) announces the continuation of the investigations started in the paper presented. In particular, correctness problems in approximate reasoning using C^T-Type-2 GMP's should be studied.

Keywords. C^T-Type-2 Fuzzy Sets on U, C^T-Type-2 General Modus Ponens(C^T-Type-2 GMP), C^T-Type-2 Semantics, Compositional Rule of Inference, Interpretation of the C^T-Type-2 GMP, Correctness of the interpreted C^T-Type-2 GMP used as Inference Rule, C^T-Type-2 Fuzzy Values, External and Internal Operations with C^T-Type-2 Fuzzy Values, Interpretation of the C^T-Type-2 GMP by using C^T-Type-2 Fuzzy Values.

1 Introduction

The notion of Type-2 Fuzzy Set was introduced in 1975 by LOFTI A. ZADEH [20]. Since then a lot of papers were published, for instance [2–5, 9–13, 15–18]. The book [9] gives a comprehensive survey of the state of the art in this field, including a long list of references.

But we think that from the standpoint of systematic theoretical-mathematical investigations of Type-2 Fuzzy Sets several questions are still open.

Before we shall discuss these questions we recall some mathematical notions and notations. Let X, Y, and Z be usual crisp sets. The classical Cartesian product of X and Y and the usual power set of X are denoted by $X \times Y$ and $\mathbb{P}X$, respectively. By $f : X \to Y$ we denote that f is a unique mapping form X into Y. The set of all mappings $f : X \to Y$ is designated by Y^X. Assume $\varphi : X \times Y \to Z$. The for fixed $y \in Y$ the function $f_y : X \to Z$ defined for every $x \in X$ by $f_y(x) =_{def} \varphi(x, y)$ is denoted by $\lambda x \varphi(x, y)$. (Denotion of the λ-calculus.) $\mathbb{R}$ and $\langle 0, 1 \rangle$ are notations for the set of all real numbers and the set of all real numbers r with $0 \leq r \leq 1$, respectively. In the following of this paper n is an integer with $n \geq 1$. X^n means the n-fold Cartesian product of X. The empty set is denoted by $\emptyset$.

Let U, C, T be non-empty sets. Generalizing the classical definition of a Type-2 Fuzzy Set formulated in [20] by ZADEH we define (see [17])

Definition 1.1. *Φ is said to be a C^T-Type-2 Fuzzy Set on U* $=_{def}$ $\Phi : U \to C^T$.

For illustrating such kind of fuzzy sets we interpret

U as a set of statements,

C as a set of certainty degrees, and

T as a set of truth values.

If we have $u \in U$, $c \in C$, $t \in T$, and $\xi \in C^T$ then the equations $\Phi(u) = \xi$ and $\xi(t) = c$ can be interpreted by "It is certain to the degree c that the statement u has the truth value t." See also [7].

2 On the Generalized Modus Ponens using Type-2 Fuzzy Sets

The Generalized Modus Ponens (GMP) introduced in 1973 by ZADEH [19] is a well-known inference rule in the field of (fuzzy logic based) approximate reasoning. Note that in this paper we shall not discuss other inference rules used in approximate reasoning.

Let Φ, Ψ, Φ', and Ψ' be C^T-Type-2 Fuzzy Sets on U, i. e. $\Phi, \Psi, \Phi', \Psi : U \to C^T$. The scheme

$$\begin{array}{ccc} \Phi & \Rightarrow & \Psi \\ \Phi' & & \\ \hline & & \Psi' \end{array}$$

is called C^T-Type-2 Generalized Modus Ponens and, for short, is denoted by C^T-Type-2 GMP.

We recall the insight that this inference rule can be meaningfully used, both in theoretical investigations and practical applications, only if this scheme is interpreted by a given semantics.

So we define [17]

Definition 2.1. *$SEM^2 = [C^T, \Pi^2, \mathcal{K}^2, Q^2]$ is said to be a C^T-Type-2 Semantics for the C^T-Type-2 GMP* $=_{def}$ *1.* $\Pi^2, \mathcal{K}^2 : C^T \times C^T \to C^T$

2. $Q^2 : \mathbb{P}C^T \to C^T$

On the basis of a given semantics SEM^2 we define how Ψ' can be computed using the fuzzy sets Φ, Ψ, and Φ' and using an interpretation of $\Rightarrow$. To do this we apply the well-known Compositional Rule of Inference introduced by ZADEH in [19]. We underline that meanwhile other methods to interpret the generalized modus ponens have been developed but due to restricted space we can not discuss these problem here.

Definition 2.2. *Assume* $u, v \in U$.

1. $REL(\Pi^2, \Phi, \Psi)(u, v) =_{def} \Pi^2(\Phi(u), \Psi(v))$
2. $\Psi'(v) =_{def} Q^2\{\mathcal{K}^2(\Phi'(u), REL(\Pi^2, \Phi, \Psi)(u, v)) \mid u \in U\}$
3. $FUNK^{SEM^2}(\Phi, \Psi, \Phi') =_{def} \Psi'$.

Obviously, $REL(\Pi^2, \Phi, \Psi)$ is a binary relation with their values in C^T, for short, it is called a binary C^T-Type-2 Relation on U. Note that by item 3 of this definition a functional operator $FUNK^{SEM^2} : \mathbb{P}C^T \times \mathbb{P}C^T \to \mathbb{P}C^T$ is defined.

This operator can be applied to define several versions of correctness which play a very important role by using the (interpreted) GMP as inference rule.

Definition 2.3.

1. *The* C^T*-Type-2 GMP is said to be correct for* Φ *and* Ψ *with respect to the semantics* $SEM^2 =_{def} FUNK^{SEM^2}(\Phi, \Psi, \Phi) = \Psi$.
2. *The* C^T*-Type-2 GMP is said to be correct with respect to the semantics* $SEM^2 =_{def}$ *For every* $\Phi, \Psi \in (C^T)^U$, $FUNK^{SEM^2}(\Phi, \Psi, \Phi) = \Psi$.

Note. Further correctness definitions are formulated in [6, 17], in particular, using a topology defined in $(C^T)^U$ and the concept of continuity.

Now, we are going to prove two correctness theorems, as examples. Therefore we assume that C is a lattice with respect to the binary relation $\leq$ on C. For $c, d \in C$ by $inf(c, d)$ we denote the infimum of c and d with respect to $\leq$. For $\xi, \eta \in C^T$ we define $\xi \precsim \eta =_{def} \forall t(t \in T \to \xi(t) \leq \eta(t))$.

Then it is well-known that C^T is a lattice with respect to $\precsim$. The infimum of ξ and η with respect to $\precsim$ is denoted by $Inf(\xi, \eta)$.

Theorem 1. *If*

1. $\forall\xi\forall\eta(\xi, \eta \in C^T \to \Pi^2(\xi, \eta) \precsim Inf(\xi, \eta))$
2. $\forall\xi\forall\eta(\xi, \eta \in C^T \to \mathcal{K}^2(\xi, \eta) \precsim Inf(\xi, \eta))$
3. $\forall X(X \subseteq C^T \wedge X \neq \Phi \wedge \forall\xi(\xi \in X \to \xi \precsim \eta) \to Q^2X \precsim \eta)$,

then for every $v \in U$, $FUNK^{SEM^2}(\Phi, \Psi, \Phi)(v) \precsim \Psi(v)$.

Proof. Assume $u, v \in U$. By assumption 1 we get

$$\Pi^2(\Phi(u), \Psi(v)) \precsim Inf(\Phi(u), \Psi(v)), \tag{1}$$

furthermore, we obtain

$$\mathcal{K}^2(\Phi(u), \Pi^2(\Phi(u), \Psi(v))) \precsim Inf(\Phi(u), \Pi^2(\Phi(u), \Psi(v))) \tag{2}$$

by assumption 2, hence by (1) and because *Inf* is $\precsim$-monotone

$$\precsim Inf(\Phi(u), Inf(\Phi(u), \Psi(v)))$$

hence by properties of *Inf*,

$$\begin{aligned} &\precsim Inf(\Phi(u), \Psi(v)) \\ &\precsim \Psi(v), \end{aligned}$$

hence by assumption 3

$$FUNK^{SEM^2}(\Phi, \Psi, \Phi)(v) =_{def} Q\{\mathcal{K}^2(\Phi(u), \Pi^2(\Phi(u), \Psi(v)))|u \in U\} \precsim \Psi(v).$$

For formulating and proving the next theorem we need the following result and notation.

Assume that the lattice $[C, \leq]$ has the unit element 1. For $t \in T$ we define $\mathbf{1}(t) =_{def} 1$. Then $\mathbf{1}$ is the unit element of the lattice $[C^T, \precsim]$.

For $f : C^T \to C^T$ we say that f is said to be Q^2-continuous
$=_{def} \forall X(X \subseteq C^T \wedge X \neq \emptyset \to f(Q^2X) \precsim Q^2\{f(x)|x \in X\})$

Theorem 2. *If at least one of the following three assumptions is satisfied*
assumption 1: (a) $\forall\xi\forall X(\xi \in X \wedge X \subseteq C^T \wedge X \neq \emptyset \to \xi \precsim Q^2X)$
(b) $Q^2\{\Psi(u)|u \in U\} \precsim Q^2\{\Phi(u)|u \in U\}$
(c) *Inf is Q^2-continuous in its first argument*
(d) $\Pi^2 = \mathcal{K}^2 = Inf$

assumption 2: (a) $\forall\eta(\eta \in C^T \to \lambda\xi\mathcal{K}^2(\xi, \Pi^2(\xi, \eta)))$ *is Q^2-continuous*
(b) $\forall\eta(\eta \in C^T \to \Pi^2(\mathbf{1}, \eta) = \mathcal{K}^2(\mathbf{1}, \eta) = \eta)$
(c) $Q^2\{\Phi(u)|u \in U\} = \mathbf{1}$

assumption 3: (a) $\forall\xi\forall X(\xi \in X \wedge X \subseteq C^T \wedge X \neq \emptyset \to \xi \precsim Q^2X)$
(b) $\exists u(u \in U \wedge \Phi(u) = \mathbf{1}$
(c) $\forall\eta(\eta \in C^T \to \Pi^2(\mathbf{1}, \eta) = \mathcal{K}^2(\mathbf{1}, \eta) = \eta$

then for every $v \in U$,

$$\Psi(v) \precsim FUNK^{SEM^2}(\Phi, \Psi, \Phi(v))$$

Proof.

ad assumption 1: By 1a, $v \in V$, and $\Psi(v) \in \{\Psi(u)|u \in U\}$, we get

$$\Psi(v) \precsim Q^2\{\Psi(u)|u \in U\}, \tag{4}$$

hence by properties of *Inf*

$$\Psi(v) \precsim Inf(Q^2\{\Psi(u)|u \in U\}, \Psi(v)), \tag{5}$$

so by 1b and monotonicity of *Inf*

$$\Psi(v) \precsim \mathit{Inf}(Q^2\{\Phi(u) | u \in U\}, \Psi(v)). \tag{6}$$

By 1c we know that *Inf* is Q^2-continuous in its first argument, hence

$$\Psi(v) \precsim Q^2\{\mathit{Inf}(\Phi(u), \Psi(v)) | u \in U\}. \tag{7}$$

The properties of *Inf* ensure

$$\Psi(v) \precsim Q^2\{\mathit{Inf}(\Phi(u), \mathit{Inf}(\Phi(u), \Psi(v))) | u \in U\}. \tag{8}$$

Assumption 1d implies

$$\Phi(v) \precsim Q^2\{\mathcal{K}^2(\Phi(u), \Pi^2(\Phi(u), \Psi(v))) | u \in U\}, \tag{9}$$

i. e.

$$\Psi(v) \precsim FUNK^{SEM^2}(\Phi, \Psi, \Phi)(v) \tag{10}$$

ad assumption 2: By 2a we get

$$\begin{aligned} & \mathcal{K}^2(Q^2\{\Phi(u) | u \in U\}, \Pi^2(Q^2\{\Phi(u) | u \in U\}, \Psi(v))) \\ \precsim\ & Q^2\{\mathcal{K}^2(\Phi(u), \Pi^2(\Phi(u), \Psi(v))) | u \in U\}. \end{aligned} \tag{11}$$

Applying 2b and 2c we obtain successively

$$\begin{aligned} & \mathcal{K}^2(Q^2\{\Phi(u) | u \in U\}, \Pi^2(Q^2\{\Phi(u) | u \in U\}, \Psi(v))) \\ =\ & \mathcal{K}^2(\mathbf{1}, \Pi^2(Q^2\{\Phi(u) | u \in U\}, \Psi(v))) \\ =\ & \mathcal{K}^2(\mathbf{1}, \Pi^2(\mathbf{1}, \Psi(v))) \\ =\ & \Pi^2(\mathbf{1}, \Psi(v)) \\ =\ & \Psi(v), \end{aligned} \tag{12}$$

hence by (11)

$$\Phi(v) \precsim Q^2\{\mathcal{K}^2(\Phi(u), \Pi^2(\Phi(u), \Psi(v))), \tag{13}$$

i. e.

$$\Phi(v) \precsim FUNK^{SEM^2}(\Phi, \Psi, \Phi)(v). \tag{14}$$

ad assumption 3: By definition of $FUNK^{SEM^2}(\Phi, \Psi, \Phi)(v)$ we have to show that

$$\text{for every } v \in U, \Psi(v) \precsim Q^2\{\mathcal{K}^2(\Phi(u), \Pi^2(\Phi(u), \Psi(v))) | u \in U\}, \tag{15}$$

hence by 3a it is sufficient to show that

$$\text{there exists an } u \in U \text{ such that } \Psi(v) \precsim \mathcal{K}^2(\Phi(u), \Pi^2(\Phi(u), \Psi(v))). \tag{16}$$

By assumption 3b we get

$$\text{there exists an } u_0 \in U \text{ such that } \Phi(u_0) = \mathbf{1} \tag{17}$$

Put $u =_{def} u_0$ in (16). Then by assumption 3c we get successively

$$\begin{aligned} & \mathcal{K}^2(\Phi(u_0), \Pi^2(\Phi(u_0), \Psi(v))) \\ =\ & \mathcal{K}^2(\mathbf{1}, \Pi^2(\mathbf{1}, \Psi(v))) \\ =\ & \Pi^2(\mathbf{1}, \Psi(v)) \\ =\ & \Psi(v) \end{aligned} \tag{18}$$

hence (15) holds.

3 On Operations with C^T-Type-2 Fuzzy Values

For getting a suitable terminology, in the following elements $\xi, \eta \in C^T$ are called C^T-Type-2 Fuzzy Values.

Consider an n-ary operation $OPER^n$ with C^T-Type-2 Fuzzy Sets $\Phi_1, \dots, \Phi_n$ on U, i. e. $OPER^n(\Phi_1, \dots, \Phi_n) : U \to C^T$.

Now, we reduce the operation $OPER^n$ to an n-ary operation $Oper^n$ with C^T-Type-2 Fuzzy Values, i. e. using the operation $oper^n : (C^T)^n \to C^T$ we define the operation $OPER^n$ as follows where $u \in U$.

$$OPER^n(\Phi_1, \dots, \Phi_n)(u) =_{def} Oper^n(\Phi_1(u), \dots, \Phi_n(u)).$$

Obviously, this reduction is used in definition 2.1 and 2.2 with respect to the binary mappings $\Pi^2, \mathcal{K}^2 : C^T \times C^T \to C^T$.

In [16] we have described two approaches to operate with context dependent fuzzy sets, the so-called external and internal operations, respectively. These ideas are adopted here for operating with C^T-Type-2 Fuzzy Values.

Definition 3.1 (External Operations with C^T-Type-2 Fuzzy Values).
Assume we have an n-ary mapping $op_e^n : C^n \to C$ called external operation. Then we "lift" this operation to an n-ary "external defined" operation Op_e^n on C^T, i. e. $Op_e^n : (C^T)^n \to C^T$ as follows. For every $\xi_1, \dots, \xi_n \in C^T$ and every $t \in T$ we put

$$Op_e^n(\xi_1, \dots, \xi_n)(t) =_{def} op_e^n(\xi(t), \dots, \xi_n(t))$$

Definition 3.2 (Internal Operations with C^T-Type-2 Fuzzy Values).
Let op_i^n be an n-ary mapping $op_i^n : T^n \to T$ called internal operation.

Then using the Extension Principle introduced by ZADEH *in [19] we "lift" this operation to an n-ary "internal defined" operation Op_i^n on C^T, i. e. $Op_i^n : (C^T)^n \to C^T$ as follows. For every $\xi_1, \dots, \xi_n \in C^T$ and every $t \in T$ we put*

$$Op_i^n(\xi_1, \dots, \xi_n)(t) =_{def}$$
$$Sup\{min(\xi_1(t), \dots, \xi_n(t)) | t_1, \dots, t_n \in T \wedge t = op_i^n(t_1, \dots, t_n)\}$$

Obviously this definition only makes sense if $C = \langle 0, 1 \rangle$, for instance. Note that this "lifting procedure" is used in the papers [10, 11] in order to define algebraic structures in $\langle 0, 1 \rangle^{\langle 0,1 \rangle}$ by "internal lifting" of $max(r, s), min(r, s), neg(r) (r, s \in \langle 0, 1 \rangle)$ and other norms, respectively.

If we assume that C is a complete lattice with the supremum operator Sup and the infimum operator Inf then we can generalize the definition of Op_i^n above as follows

$$Op_i^n(\xi_1, \dots, \xi_n)(t) =_{def}$$
$$Sup\{Inf(\xi_1(t), \dots, \xi_n(t)) | t_1, \dots, t_n \in T \wedge t = op_i^n(t_1, \dots, t_n)\}$$

For our investigations it is favourable to introduce the following further generalization of the extension principle where T and C are arbitrary non-empty sets.

Assume $\mu_i^n : C^n \to C$ and $Q_i : \mathbb{P}C \to C$ then we put

$$Op_{i,\mu_i^n,Q_i}^n(\xi_1, \dots, \xi_n)(t) =_{def}$$
$$Q_i\{\mu_i^n(\xi_1(t), \dots, \xi_n(t)) | t_1, \dots, t_n \in T \wedge t = op_i^n(t_1, \dots, t_n)\}$$

4 On Four Kinds of Interpreting the C^T-Type-2 Generalized Modus Ponens

In section 3 we have reduced general operations with C^T-Type-2 Fuzzy Sets to operations with their C^T-Type-2 Fuzzy Values. Note that this reduction principle is already applied in section 2 to interpret the C^T-Type-2 *GMP*. Furthermore, in section 3 we have stated that for operating with C^T-Type-2 Fuzzy Values there are two methods, the external and the internal lifting.

We recall that for interpreting the C^T-Type-2 *GMP* we have two mappings.

$$\Pi^2, \mathcal{K}^2 : C^T \times C^T \to C^T$$

for operating with C^T-Type-2 Fuzzy Values. Then we have defined (see definition 2.2) two binary C^T-Type-2 Fuzzy Relations $\lambda u \lambda v \Pi^2(\Phi(u), \Psi(v))$ and $\lambda u \lambda v \mathcal{K}^2(\Phi'(u), \Pi^2(\Phi(u), \Psi(v))))$.

Despite other possibilities not discussed here, referring to section 3 each of the functions Π^2 and $\mathcal{K}^2$ can be constructed by an external and an internal lifting from C to C^T and from T to C^T, respectively.

So using this "lifting philosophy" we obtain the following four cases of interpreting the C^T-Type-2 *GMP* where in the following we generally assume $u, v \in U$ and $t \in T$.

Case 1. The EE-Interpretation where "EE" means "external-external".

In this case we assume Π^2 and $\mathcal{K}^2$ are generated by external lifting of the functions $\pi_e, \kappa_e : C \times C \to C$ from C to C^T. Therefore we define

Definition 4.1. $SEM_{ee} = [\pi_e, \kappa_e, Q]$ *is said to be an EE-Semantics for the* C^T*-Type-2 GMP* $=_{def}$ *1.* $\pi_e, \kappa_e : C \times C \to C$

2. $Q : \mathbb{P}C \to C$

Using this semantics we define the so-called EE-Interpretation of the C^T-Type-2 *GMP* as follows

Definition 4.2.

1. $REL_e(\pi_e, \Phi, \Psi)(u, v)(t) =_{def} \pi_e(\Phi(u)(t), \Psi(v)(t))$

2. $\Psi'_{ee}(v)(t) =_{def} Q\{\kappa_e(\Phi'(u)(t), REL_e(\pi_e, \Phi, \Psi)(u, v)(t)) | u \in U\}$

3. $FUNK^{SEM_{ee}}(\Phi, \Psi, \Phi') =_{def} \Psi'_{ee}$

Case 2. The EI-Interpretation where "EI" means "external-internal".

In this case we assume that Π^2 is generated by external lifting using the external function $\pi_e : C \times C \to C$ and $\mathcal{K}^2$ is generated by internal lifting using the internal function $\kappa_i : T \times T \to T$.

Definition 4.3. *$SEM_{ei} = [\pi_e, \kappa_i, \mu_i, Q_i, Q]$ is said to be an EI-Semantics for the C^T-Type-2 GMP*

$=_{def}$ 1. $\pi_e : C \times C \to C$

2. $\kappa_i : T \times T \to T$

3. $\mu_i : C \times C \to C$

4. $Q_i : \mathbb{P}C \to C$

5. $Q : \mathbb{P}C \to C$

Using this semantics we define the so-called EI-Interpretation of the C^T-Type-2 *GMP* as follows

Definition 4.4.

1. $REL_e(\pi_e, \Phi, \Psi)(u,v)(t) =_{def} \pi_e(\Phi(u)(t), \Psi(v)(t))$
2. $\mathcal{K}^2(\Phi'(u), REL_e(\pi_e, \Phi, \Psi)(u,v)(t)) =_{def}$ $Q_i\{\mu_i(\Phi'(u)(t_1), REL_e(\pi_e, \Phi, \Psi)(u,v)(t_2)) | t_1, t_2 \in T \wedge t = \kappa_i(t_1, t_2)\}$
3. $\Psi'_{ei}(v)(t) =_{def} Q\{\mathcal{K}^2(\Phi'(u), REL_e(\pi_e; \Phi, \Psi)(u,v)(t)) | u \in U\}$
4. $FUNK^{SEM_{ei}}(\Phi, \Psi, \Phi') =_{def} \Psi'_{ei}$

Case 3. The IE-Interpretation where "IE" means "internal-external".

In this case we assume that Π^2 is generated by internal lifting using the internal function $\pi_i : T \times T \to T$ and $\mathcal{K}^2$ is generated by external lifting using the external function $\kappa_e : C \times C \to C$.

Definition 4.5. *$SEM_{ie} = [\pi_i, \mu_i, Q_i, \kappa_e, Q]$ is said to be an IE-Semantics for the C^T-Type-2 GMP*

$=_{def}$ 1. $\pi_i : T \times T \to T$

2. $\mu_i : C \times C \to C$

3. $Q_i : \mathbb{P}C \to C$

4. $\kappa_e : C \times C \to C$

5. $Q : \mathbb{P}C \to C$

Using this semantics we define the so-called IE-Interpretation of the C^T-Type-2 *GMP* as follows

Definition 4.6.

1. $REL_i(\pi_i, \mu_i, Q_i, \Phi, \Psi)(u,v)(t) =_{def}$ $Q_i\{\mu_i(\Phi(u)(t_1), \Psi(v)(t_2)) | t_1, t_2 \in T \wedge t = \pi_i(t_1, t_2)\}$
2. $\Psi'_{ie}(v)(t) =_{def} Q\{\kappa_e(\Phi'(u)(t), REL_i(\pi_i, \mu_i, Q_i, \Phi, \Psi)(u,v)(t)) | U \in U\}$
3. $FUNK^{SEM_{ie}}(\Phi, \Psi, \Phi') =_{def} \Psi'_{ie}$

Case 4. The II-Interpretation where "II" means "internal-internal".

In this case we assume that Π^2 and $\mathcal{K}^2$ are generated by an internal function $\pi_i : T \times T \to T$ and an internal function $\kappa_i : T \times T \to T$, respectively.

Definition 4.7. *$SEM_{ii} = [\pi_i, \mu_i, Q_i, \kappa_i, \mu_i', Q_i', Q]$ is said to be an II-Semantics for the C^T-Type-2 GMP*

$=_{def}$ *1.* $\pi_i : T \times T \to T$

2. $\mu_i : C \times C \to C$

3. $Q_i : \mathbb{P}C \to C$

4. $\kappa_i : C \times C \to C$

5. $\mu_i' : C \times C \to C$

6. $Q_i' : \mathbb{P}C \to C$

7. $Q : \mathbb{P}C \to C$

Using this semantics we define the so-called II-Interpretation of the C^T-Type-2 GMP as follows

Definition 4.8.

1. $REL_i(\pi_i, \mu_i, Q_i, \Phi, \Psi)(u, v)(t) =_{def}$
$Q_i\{\mu_i(\Phi(u)(t_1), \Psi(v)(t_2)) | t_1, t_2 \in T \wedge t = \pi_i(t_1, t_2)\}$

2. $\mathcal{K}^2(\Phi'(u), REL_i(\pi_i, \mu_i, Q_i, \Phi, \Psi)(u, v)(t')) =_{def}$
$Q_i'\{\mu_2'(\Phi'(u)(t_1'), REL_i(\pi_i, \mu_i, Q_i, \Phi, \Psi)(u, v)(t_2')) | t_1', t_2' \in T \wedge t' = \kappa_i(t_1', t_2')\}$

3. $\Psi_{ii}'(v)(t') =_{def} Q\{\mathcal{K}^2(\Phi'(u), REL_i(\pi_i, \mu_i, Q_i, \Phi, \Psi)(u, v)(t')) | u \in U\}$

4. $FUNKT^{SEM_{ii}}(\Phi, \Psi, \Phi') =_{def} \Psi_{ii}'$

5 Conclusions

With respect to the concepts developed in section 4 we are faced with the problem to investigate and to apply a Type-2 approximate reasoning based on the four kinds of the interpreted C^T-Type-2 Generalized Modus Ponens.

This will be done in a forthcoming paper because of restricted space here.

Acknowledgement

The author would like to thank VOLKHER KASCHLUN for his help in preparing the manuscript.

References

[1] R. E. Bellmann and L. A. Zadeh. Local and fuzzy logics. In J. M. Dunn and G. Epstein, editors, *Modern Uses of Multiple-Valued Logic — Invited Papers of 5th ISMVL Symposium 1975*, pages 103–165. Reidel, Dordrecht, 1977.

[2] D. Dubois and H. Prade. Operations in a fuzzy-valued logic. *Information and Control*, 43(2):224–240, November 1979.

[3] R. I. John. Type 2 fuzzy sets: an appraisal of theory and applications. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(6):563–576, Dec. 1998.

[4] N. N. Karnik and J. M. Mendel. Operations on type-2 fuzzy sets. *Fuzzy Sets and Systems*. to appear.

[5] N. N. Karnik, J. M. Mendel, and Q. Liang. Type-2 fuzzy logic systems. *IEEE-FS*, 7(6):643, December 1999.

[6] St. Lehmke, B. Reusch, K.-H. Temme, and H. Thiele. On interpreting fuzzy if-then rule bases by concepts of functional analysis. Technical Report CI-19/98, University of Dortmund, Collaborative Research Center 531, February 1998.

[7] St. Lehmke Logics which allow Degrees of Truth and Degrees of Validity - A way of handling Graded Truth Assessment and Graded Trust Assessment within a single framework. Ph.–D. Thesis. University of Dortmund (Germany). Department of Computer Science I. October 2001

[8] J.M. Mendel. Computing with words when words can mean different things to different people. In *3rd Annual Symposium on Fuzzy Logic and Applications*, Rochester, New York, USA, June 22-25 1999.

[9] J.M. Mendel. Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice Hall PTR. 2001, 555 pages

[10] M. Mizumoto and K. Tanaka. Some properties of fuzzy sets of type 2. *Information and Control*, 31(4):312–340, August 1976.

[11] M. Mizumoto and K. Tanaka. Fuzzy sets under various operations. In *4th International Congress of Cybernetics and Systems*, Amsterdam, The Netherlands, August 21-25, 1978.

[12] J. Nieminen. On the algebraic structure of fuzzy sets of type-2. *Kybernetica*, 13(4), 1977.

[13] H. Thiele. On logical systems based on fuzzy logical values. In *EUFIT '95 — Third European Congress on Intelligent Techniques and Soft Computing*, volume 1, pages 28–33, Aachen, Germany, August 28–31, 1995.

[14] H. Thiele. On the concept of qualitative fuzzy set. In *Proceedings of the Twenty-Ninth International Symposium on Multiple-Valued Logic*, pages 282–287, Freiburg, Germany, May 20–22, 1999.

[15] H. Thiele. On approximate reasoning with context-dependent fuzzy sets. In *WAC 2000 — Fourth Biannual World Automation Congress*, Wailea, Maui, Hawaii, USA, June 11–16 2000.

[16] H. Thiele. A New Approach to Type-2 Fuzzy Sets. In *the Congress of Logic Applied to Technology (LAPTEC 2001)* Sao Paulo, Brazil, November 12-14, 2001. Conference Proceedings, pages 255–262.

[17] H. Thiele. On approximative reasoning with Type-2 Fuzzy Sets. Accepted Paper. *IPMU 2002*, Annecy, France, July 1–5, 2002.

[18] M. Wagenknecht and K. Hartmann. Application of fuzzy sets of type 2 to the solution of fuzzy equation systems. *Fuzzy Sets and Systems*, 25:183–190, 1988.

[19] L. A. Zadeh. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes.. IEEE Trans. on Systems, Man, and Cybernetics, vol. SMC-3, pp. 26-44, 1973

[20] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning — I-III. *Information Sciences*, (I) 8:199–249, (II) 8:301–357 (III) 9:43–80, 1975.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Paraconsistent Knowledge for Misspelling Noise Reduction in Documents

Emerson Luís dos Santos Fabiano Mitsuo Hasegawa Bráulio Coelho Ávila
Celso Antônio Alves Kaestner

Pontifical Catholic University of Paraná, Curitiba - PR, Brazil
`{emerson, fmitsuo, avila, kaestner}@ppgia.pucpr.br`*

Abstract. There has been a large variety of search engines available on the Web. At least most of them act as regular Information Retrieval tools, where the collection is the whole World Wide Web. Although evaluations are made according to human-pointed references of the set of relevant documents in a collection for a given query, at least a bulk of works are variations of the classic logic, probabilistic and vector models, without taking memory issues — which underlies cognitive processes, specially reasoning — into account — neither the models, nor the variations use knowledge. Thus, they are supposed to retrieve pages that contain the words of a query, rather than the virtual set of pages containing the subject those words are intended to point. It sounds human-like heuristics and techniques would perform better. In this paper, some questions concerning IR are discussed, whilst an add-on process for misspelling noise reduction using paraconsistent knowledge over automatic acquisition of behaviours of features is presented. The domain is retrieval of ontologies represented in Resource Description Framework (RDF) according to keys provided by the user. This is an extremely short step into the primary improvements likely to be attained by providing search engines with semantic knowledge.

1 Introduction

Search engines are usually build according to three paradigms well defined in the Information Retrieval (IR) field: boolean, vector and probabilistic models [2]. Those techniques were developed under assumptions concerning mathematic and statistical issues. The current state of art, from the perspective of results achieved, is not bad but is far from what human beings can do and specially far from what they wish.

IR is not being attacked the way it should be. Since the target is a human-like retrieval of information, mechanisms to determine which documents must be fetched in satisfaction of a query should be closer to cognitive processes, instead of just strictly word-checking/word-counter models.

In this work, an approach for reduction of noise due to misspelling mistakes is introduced. This approach is pimarily based on knowledge about similarity of words. Overmore, in order not to allow mistaken unifications, some heuristics concerning the behaviour of some features noticed in collections are automatically acquired and used. All the terms are qualified with

---

*This work was supported by CAPES, Brazil.

those features. Terms are then refined through paraconsistent knowledge of authenticity of words, which is related to those features automatically acquired. However, this knowledge is provided by experts.

At the end of that process, noise due to misspelling mistakes is reducted. The motivation is that some collections may present lots of misspelling mistakes which can jeopardise the performance of a search engine.

Section 2 shows how naïve string approximation methods for misspelling noise tolerance can be replaced by rules conveying semantic knowledge. Next, knowledge from experts is expressed in the form of paraconsistent rules in order to avoid inadequate unification, discarding erroneous actions. The whole architecture of a search engine using the technique for noise reduction presented is provided in Section 4. Finally, a conclusion is presented. Examples of additional information used are supplied in the Appendices.

2 Knowledge Applied to Noise Tolerant Unification

As it is common in real world data, the RDF collection contained several typing errors. If no tolerance was applied, the search engine would not be able to identify *approximmation* was, in fact, intended to be *approximation*, for instance.

The tolerant unification was implemented according to some heuristics which try to represent the way human beings reason when they must decide whether two strings are or are not equal. The heuristics were expressed as rules and concern the following ideas:

- Two words are equal if, when compared, a mistake does not occur more than once before it is forgot;
- A mistake is forgot if it was not repeated within the same word in the last θ characters (in this paper, the value of θ was 3);
- There are three kinds of mistakes: *transposition* mistake — which is the reversal of two letters —, *typing* mistake — when a different letter is found instead of the expected one — and *missing* mistake — when a letter is missed[1].

It can be noticed in Table 1 that allowing smooth matching was useless for *recall*. Nonetheless, its effects for *precision* were extremely bad: lots of inadequate unifications were allowed.

Initially, this approach would be used in the search. Every term in the index which matched the terms on the query would be used to retrieve the related documents. However, since this technique proved not to be suitable because of the trouble already exposed — inadequate unifications — according to the results provided later in this section. Actually, the way it was being dealt with was completely incorrect. Consequently, a new strategy turned out to be required: it is not the possible matchings that should be recognized in the search; search must look for exact words. Instead, the index must be properly constructed in order to incorporate misspelling issues. This is explained in the next section.

---

[1] Additional letters are treated as missed letters too, as none of the words is considered to be right *a priori*.

Table 1: Comparison between regular and smooth search.

Query	Regular		Smooth	
	Precision	Recall	Precision	Recall
q_1	0.66	1	0.16	1
q_2	0.41	0.24	0.38	0.24
q_3	0.44	0.11	0.45	0.14
q_4	0.25	0.08	0.25	0.08
q_5	0	0	0	0
q_6	0.27	0.1	0.25	0.1
q_7	0	0	0	0
q_8	0.18	0.5	0.12	0.5
q_9	0.14	0.25	0.14	0.25
q_{10}	0	0	0	0
q_{11}	0.1	0.31	0.1	0.31
$\overline{q_i}$	0.22	0.24	0.17	0.24

3 Paraconsistency as a Foundation to Represent Word Validation Knowledge

The trouble with smooth search arises when an arbitrary word is compared with similar authentic words[2]. The tolerant unification was designed not to care about some differences when comparing two terms in order to allow partial matchings. Nevertheless, the tolerant unification cannot realize when two similar words are both authentic because it does not have any knowledge concerning this issue. This cannot be disregarded because it was a deficiency introduced by the tolerant unification.

Normally, a misspelled word is not found in a dictionary; however, sometimes people can transform the intended word into another word which is not invalid: the new word might have been misspelled in such a way that it still exists in a thesaurus. Plainly, authenticity of a word is not just the same as its existence in a dictionary.

The terms of the indices and their respective frequencies in the documents are all the available information to decide whether a term t_2 is either just the authentic term t_1 misspelled or, indeed, another authentic term. If it is a misspelling, its frequency should be added to t_1's and its entry removed from the index.

3.1 Notes on Paraconsistent Logics

ParaLog [1] is a reasoner based on Paraconsistent Logics [6, 3, 4] implemented on Prolog. Thus, *ParaLog* is able to represent explicitly a negation, instead of just supressing it. The Paraconsistent Logic addressed here, which is the basis for *ParaLog*, is the *Infinitely Valued Paraconsistent Logics* [11], whose lattice is showed in figure Figure 1 (the operator $\leq$ indicates the usual order of the real numbers).

Each proposition in *ParaLog* is qualified with an annotation of the lattice τ. Both evidences may be used to determinate a single discrete value of *confidence* on a proposition, from the set {`false`, `uncertain`, `true`}, using the *certainty degree* G, as defined in [6], showed in (1).

---

[2] A word is considered to be authentic if it is not a misspelling mistake.

$$\begin{aligned}
\tau &= < |\tau|, \leq> \\
|\tau| &= [0, 1] \times [0, 1] \\
\leq &= \{((\mu_1, \rho_1), (\mu_2, \rho_2)) \in \\
&\quad ([0, 1] \times [0, 1])^2 \mid \\
&\quad \mu_1 \leq \mu_2 \text{ e } \rho_1 \leq \rho_2 \\
&\}
\end{aligned}$$

Figure 1: The Infinitely Valued Bi-Evidential Lattice.

Table 2: Conversion of *certainty degree* into *confidence*.

Certainty Degree	Confidence
$G_1 \geq +0.5$	The *confidence* of p_1 is assumed to be `true`.
$G_1 \leq -0.5$	The *confidence* of p_1 is assumed to be `false`.
$-0.5 < G_1 < +0.5$	p_1 cannot be stood on safely.

$$G = \mu - \rho \tag{1}$$

Let p_1 be a proposition whose annotation $[\mu_1, \rho_1]$ yields a *certainty degree* G_1. So with a *certainty degree* threshold $\phi = 0.5$, for example, the assertions in Table 2 hold. According to Table 2, if the *confidence* of a proposition p is `uncertain`, some feature can be chosen to remove this uncertainty.

3.2 Heuristics for Word Authenticity

Some craw heuristics related to term, based on relations between *confidence* on the authenticity of an arbitrary word and behaviours of some features noticed in collections, were used to qualify each term.

The behaviours of the features were converted from continuous to discrete for simplicity. That allowed the use of facts only with annotations corresponding to values in the subset {`false`, `true`} of the lattice τ. The resulting heuristics are presented in Table 3. In this table, *opponents* of a term t are all the terms in the collection which are similar to t according to the smooth term unification.

The thresholds for a decision between the elements of the set {`false`, `true`} — that, actually, denote {$\neg$ `high`, `high`} — are domain dependent. In the tests presented in this paper, dynamic thresholds λ_k were obtained as indicated in (2) for each of the k features evaluated.

$$\lambda_k = \overline{f}_k \tag{2}$$

Table 3: Discrete heuristics to determine the authenticity of a word.

Existence of	Present	Authenticity
At least one high *tf*	+	⇈
High *df*	+	⇈
High frequency of high-*tf*s	+	⇈
High frequency of opponents	+	⇈
Term referred in *WordNet*	+	⇈
High frequency of high-*tf* opponents	+	⇊
At least one high-*tf* opponent	+	⇊

In (2), $\overline{f}_k$ means the global *average* of the values of the feature k[3].

In this approach, each term t is evaluated according to each of the features k. Thereby, a vector $V = \{v_1^1, v_1^2, \cdots, v_1^k,\ v_2^1, v_2^2, \cdots, v_2^k,\ \ldots,\ v_n^1, v_n^2, \cdots, v_n^k\}$ is obtained, where v_i^k is internally represented as a fact:

(a) either $feature\_k(t_i)^{[0,\,1]}$ if $f_k^i < \lambda_k$;

(b) or $feature\_k(t_i)^{[1,\,0]}$ otherwise ($f_k^i \geq \lambda_k$).

3.3 The Process of Word Authentication

Thus far, every term was represented by a set of facts concerning the presence of features observed in the collection, as defined in the previous subsection. In other words, there are more explicit information about terms of the collection than before.

Rules based on domain knowledge are implemented as clauses named `authentic` and annotated with any element of the lattice τ and are formed by conjunctions of queries concerning the information in Table 3 annotated only with elements of the subset {`false`, `true`} of the lattice τ. The high number of possible combinations permits paraconsistent results. Therefore, *ParaLog* is used as a query module to retrieve the evidences related to the authenticity of a given term.

Finally, both evidences — *belief* and *disbelief* — are used to get the *certainty degree* of authenticity of that term as showed in (1). Consequently, the *confidence* of a term can be obtained as in Table 2.

With knowledge of authenticity of terms available, it is possible to create a new index of *absolute frequency* corrected. Let I be the index of *absolute frequency* of all terms in the collection, where rows are terms and columns are documents. Each element $e_{i,j}$ contains the frequency of the term t_i in the document d_j. The confidence of each term t_i is evaluated: if the term t_i is authentic, then its entry in the index I is conserved; otherwise, each of its frequencies F_i^j is equally divided among all their authentic opponents O_i, increasing the frequencies $F_{O_i}^j$, and its entry is removed from the index I.

---

[3] Notice there is exactly one value of each feature for each term.

Table 4: Predicted Results × Manual Reference.

		Real Class	
		+	−
Predicted Class	+	4165	25
	−	59	27

Table 5: Automatic Authentication Accuracy Percentages Discriminated by Class

Class	Accuracy Percentages
Positive	0.9935
Negative	0.7023
Balanced General	0.8353

3.4 *Evaluation of the Authentication Process*

The results obtained with the automatic authentication of terms were compared with the results expected according to a human-made reference. Table 4 contrasts those results.

The high value of real positives mistakenly predicted as negatives are due to the high number of acronyms. Acronyms are difficult to be identified without context. Thus, they are likely to be wrongly classified as not authentic if they have a low frequency.

Words misspelled lots of times may be motivated to be considered authentic. On the other hand, if a term is misspelled and there are not enough ocurrences of its correct form to allow the identification of the authentic form, it is almost impossible to figure out the right form if it does not appear in *WordNet*.

More than one mistake inside the interval delimited by the smooth unification threshold forbids the identification of misspellings. To cope with that, a more robust method is needed.

Table 5 makes a comparison of the accuracy discriminated by class. Whereas the noise is small, the classes are not balanced — so the total accuracy rate needs to take into account that information.

4 The Architecture and the Operational Semantic of the System

This section presents the architecture of the program which implements the issues discussed along the paper. The architecture is depicted in Figure 2 and the system is explained step by step along the section.

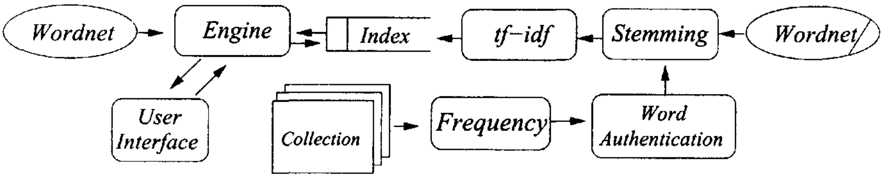


Figure 2: The system architecture.

The collection utilized in the whole set of experiments was obtained from the DAML [8] ontologies repository at `http://www.daml.org/ontologies`. Although RDF [10, 5] has a lot of tags representing properties, only two are interesting for this work: `label`, which conveys the name of a resource in natural language, and `comment`, which provides a description of a resource.

Once the collection had been downloaded, the first step was to generate indices. A list of *stop words* helped to select the words to include in the index. An *absolute frequency* index was then generated.

Secondly, using the paraconsistent knowledge rules about authenticity of words like the examples provided in Appendix A, a new index was obtained according to the procedures described in Subsection 3.3, reducing noise due to misspelled words. Example facts concerning the extra information automatically acquired about terms are provided in Appendix B.

Morphological *WordNet* [9, 7] operations were then used to reduce this new index to a normalized form, where only primitive words exist.

Finally, that last index generated so far is used to create the ultimate index: a *tf-idf* index. This is the ideal index upon which the search will be carried out. Recall that this index is normalized — only primitive forms — and noise due to misspelling errors is inhibited.

5 Conclusion

It seems fairly clear knowledge is required in IR. The targets of this community are too much related to cognitive processes because, in essence, they want to do something people use knowledge to do — memory is the basis for every cognitive process in human beings.

Disambiguation, of course, is the main point. That is where knowledge will surely surprise the crowd of researchers who persist in knowledge-free techniques. The argument is clear: people only can disambiguate because they use their knowledge structures to guide the process.

A very important point to discuss is the use of Data Mining techniques in the IR field. In Data Mining, there is usually no or little knowledge about the database. The aim is to mine the database in order to discover that knowledge. It seems weird to use Data Mining techniques for understanding in IR: the goal is not to discover knowledge; the goal is to understand texts. Experts have that knowledge. Thereby, expert knowledge should be used to understand text. In short, there is no need to look for knowledge because it is already known.

Memory is the mine to be explored. Cognitive processes are the tools which must be polished. Natural language will only be perfectly tractable when computers have the knowledge and competences human beings do.

6 Acknowledgements

We are thankful to Márcio Roberto Starke for his contribution in earlier studies of this work. This research has been partially funded by Capes, Brazil.

References

[1] B.C. Ávila. *Uma Abordagem Paraconsistente Baseada em Lógica Evidencial para Tratar Exceções em Sistemas de Frames com Múltipla Herança*. PhD thesis, Escola Politécnica da Universidade de São Paulo, São Paulo, 1996.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* ACM Press, New York, 1999.

[3] H.A. Blair and V.S. Subrahmanian. Paraconsistent logic programming. In *Proc. 7th Conference on Foundations of Software Technology and Theoretical Computer Science, Lecture Notes on Computer Science*, volume 287, pages 340–360. Springer-Verlag, 1987.

[4] H.A. Blair and V.S. Subrahmanian. Paraconsistent foundations of logic programming. *Journal of Non-Classical Logic*, 5(2):45–73, 1988.

[5] D. Brickley and R.V. Guha. *Resource Description Framework (RDF) Schema Specification.* W3C World Wide Web Consortium, March 2000. W3C Candidate Recommendation (Work in progress).

[6] N.C.A. da Costa et al. *Lógica Paraconsistente Aplicada.* Atlas, São Paulo, 1999.

[7] G.A. Miller et al. *Five Papers on WordNet.* CLS Report 43, Cognitive Science Laboratory, Princeton University, 1990.

[8] P.F. Patel-Schneider F. van Harmelen and I. Horrocks. *Reference Description of the DAML+OIL (March 2001) — Ontology Markup Language.* Available on the Internet at `http://www.daml.org/2001/03/reference.html` on Feb 19th 2002, March 2001. Work in progress.

[9] C. Fellbaum, editor. *WordNet: an electronic lexical database.* The MIT Press, Cambridge, Massachusetts, 1998.

[10] O. Lassila and R.R. Swick. *Resource Description Framework (RDF) Model and Syntax.* W3C World Wide Web Consortium, February 1999. W3C Recommendation.

[11] V.S. Subrahmanian. Towards a theory of evidential reasoning in logic programming. In *Logic Colloquium '87, The European Summer Meeting of the Association for Symbolic Logic*, Granada, Spain, July 1987.

A Paraconsistent Rules used in Word Validation

Some of the rules upon which a *ParaLog* query is triggered.

```
...
authentic(T):[0.9, 0] <--
        wordnet(T):[1,0] &
        high_df(T):[1,0].
authentic(T):[0, 0.7] <--
        high_tf_opponent(T):[1, 0] &
        high_freq_high_tf_opponents(T):[1, 0].
...
```

B A Piece of the Information Automatically Acquired

Some facts representing the information automatically acquired about terms of the collection are supplied below as examples. There is a clause for each pair term-feature.

```
...
wordnet(ontology):[1, 0].
high_tf(person):[0, 1].
high_df(person):[1, 0].
...
```

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Automata with concurrency relations
– a survey –

Manfred Droste[1] Dietrich Kuske[2]

Abstract. Automata with concurrency relations, which occurred in the verification methods of concurrent programs, are labeled transition systems with state-dependent concurrency relations on the actions. They generalize the asynchronous transition systems and trace alphabets. Here we survey results obtained on such automata with connections to domain theory, Petri nets, graph-theoretic representations and algebraic and logical definability of languages.

1 Introduction

In formal verification of concurrent reactive systems, state-space exploration techniques and tools developed on their basis are becoming increasingly established. Their main limit, however, is the excessive size of the global state space. Recently, it has been proposed to exploit the possible independency of actions to consider only part of the global state space [43, 37, 35, 36]. These partial order methods have been implemented [36] and used for formal validation of industrial concurrent programs [36]. For a survey on this, see *e.g.* [35]. For the mentioned partial order methods, the underlying system is modelled as a finite-state automaton together with a relation indicating when two actions can operate independently of each other in a given state. This independence may differ from state to state. Considering such conditional independencies may decrease memory requirements for the verification of real-size protocols effectively as shown *e.g.* in [37].

It is the aim of this survey to describe an automaton model suitable for dealing with state-dependent independencies and to outline results obtained in conncetions with domain theory, Petri nets, and formal language theory. This automaton model actually arose independently through a different background which we would like to describe next. Petri nets provide a well investigated model for concurrent processes. The dynamic behaviour of such nets, in particular condition/event-systems, led Mazurkiewicz [53, 54] to the introduction of *trace alphabets* and their associated *trace monoids*. Independently and earlier, these monoids had also arisen in combinatorial questions on rearrangement [12]. *Trace theory* now has a well developed mathematical theory, see [16, 18]. One considers pairs (Σ, I) where Σ is the set of actions, and I a symmetric and irreflexive binary relation on Σ describing the independence of two actions. The trace monoid or free partially commutative monoid is then defined as the quotient $\Sigma^*/{\sim}$ where $\sim$ is the congruence on the free monoid Σ^* generated by all pairs (ab, ba) with $(a, b) \in I$. Consequently, in this model the independence of two actions is given by a single global relation, actions are always executable, and it abstracts away from the possibly different states of an underlying system. The automata with concurrency relations presented here are a much refined model, taking into account the states of the system, the transitions, and a collection of local independence relations.

---

[1] Institut für Algebra, Technische Universität Dresden, D–01062 Dresden, Germany, email: droste@math.tu-dresden.de

[2] Department of Mathematics and Computer Science, University of Leicester, Leicester, LE1 7RH, UK, email: d.kuske@mcs.le.ac.uk

An *automaton with concurrency relations* is a quadruple $\mathcal{A} = (Q, \Sigma, T, \|)$. Here, Q is the set of states or situations, Σ is the set of actions, $T \subseteq Q \times \Sigma \times Q$ is the set of transitions, and $\| = (\|_q)_{q \in Q}$ is a collection of concurrency relations $\|_q$ for Σ, indexed by the possible states $q \in Q$. The relation $a \|_q b$ captures that the actions a and b can occur independently of each other in state or situation q. Observe that possibly $\|_p \neq \|_q$ if $p \neq q$. Hence such automata with concurrency relations generalize the asynchronous transition systems and trace automata which were investigated by various authors, *e.g.* [2, 65, 67, 60, 5, 72], and used to provide a semantics for CCS [56, 7], to model properties of computations in term rewriting systems and in the lambda calculus [1, 6, 3, 40, 52], and in dataflow networks [60]. Structures with varying independence relations have also been investigated in [42]. Note that trace alphabets arise simply by considering automata with concurrency relations $\mathcal{A}$ with just a single state. In general, let $\mathrm{CS}(\mathcal{A})$ comprise all finite computation sequences of $\mathcal{A}$, with concatenation as (partially defined) monoid operation. Similarly as in trace theory, we declare two computation sequences of the form $(p, a, q)(q, b, r)$ and $(p, b, q')(q', a, r)$ equivalent, if $a \|_p b$. This generates a congruence $\sim$ on $\mathrm{CS}(\mathcal{A})$, and the quotient $\mathrm{M}(\mathcal{A}) := \mathrm{CS}(\mathcal{A})/{\sim} \cup \{0\}$ (formally supplemented with 0 to obtain an everywhere defined monoid operation) is called the *concurrency monoid* associated with $\mathcal{A}$.

Deterministic automata with concurrency relations where introduced in [19, 20], where their associated domains of computations where investigated. These arise as follows. The prefix order for computation sequences induces a preorder $\lesssim$ and an equivalence relation $\sim$ on the set $\mathrm{CS}^\infty(\mathcal{A})$ of all (finite or infinite) computation sequences of $\mathcal{A}$ and hence a poset $\mathrm{M}^\infty(\mathcal{A}) := (\mathrm{CS}^\infty(\mathcal{A}), \lesssim)/{\sim}$. Let $\star \in Q$ be a fixed state, and now the poset $(D(\mathcal{A}), \leq)$ comprising all elements of $\mathrm{M}^\infty(\mathcal{A})$ whose representing computation sequences start in $\star$ is called the *domain of computations* associated with $\mathcal{A}$. They can also be defined if $\mathcal{A}$ is nondeterministic, and their order structure has been completely described in [19, 20, 44]. Under suitable assumptions on $\mathcal{A}$, they are closely related with various classes of domains arising in denotational semantics of programming languages like L-domains and event domains. In particular, stably concurrent automata generate precisely the class of all dI-domains.

In place/transition-nets, where places may carry several tokens and transitions may fire with multiplicities, the independence of two actions depends on the underlying marking, *i.e.* the existence of sufficiently many tokens (resources of the modelled system). Hence these nets provide natural structures whose dynamic behaviour can be modelled by automata with concurrency relations. In [29, 30], this relationship was described by an adjunction between a category of automata with concurrency relations and a category of Petri nets, similarly to the results of [58, 72, 57].

In trace theory, an important tool used in many difficult applications is the representation of traces by labeled graphs, or even labeled partially ordered sets. Here we present such a representation for the computations in $\mathrm{M}(\mathcal{A})$ [9]. This assumes that $\mathcal{A}$ is stably concurrent, *i.e.* that the concurrency relations of $\mathcal{A}$ depend locally on each other in the form of the cube and the inverse cube axiom. Recall that stably concurrent automata generate precisely the dI-domains; these distributivity properties are crucial in the proof. We also note that somewhat surprisingly, they are closely related with Levi's lemma in trace theory. Very recently, we learned that similar properties occurred in the area of Gaussian and of Garside monoids [14, 15, 62].

It seems that many results on trace theory can be generalized to concurrency monoids $\mathrm{M}(\mathcal{A})$, provided the right assumptions on $\mathcal{A}$ have been found. We illustrate this by results from formal language theory. Also, this presents borderlines to the problem how far these classical results can be extended to more general monoids.

Kleene's classical description of the recognizable languages in a free monoid as the rational languages was generalized by Ochmański [59] to trace monoids. Under suitable assumption on $\mathcal{A}$, both directions of this result have been extended to languages in $\mathrm{M}(\mathcal{A})$ [21]. Furthermore, in [27], we define divisibility monoids and show a related result on recognizable languages in these monoids. Using these results, some of the statements from [21] can be sharpened. Thus, we obtain descriptions of the recognizable languages in terms of certain rational expressions, homomorphisms into finite monoids, and M-automata. Asynchronous-cellular automata for traces are a model with distributed control. They are capable of accepting precisely all recognizable trace languages [73]. In [23] this model has been extended to arbitrary pomsets without autoconcurrency and it was shown that in particular all recognizable languages in the concurrency monoid can be accepted by an asynchronous-cellular automaton. Moreover, since we can represent computations in a unique way by labeled posets, we can use logical languages to describe properties of these posets and thus define languages of computations satisfying given formulas. We show that if $\mathcal{A}$ is finite and stably concurrent, then a language in $\mathrm{M}(\mathcal{A})$ is recognizable iff it is definable by a formula of monadic second order logic [24]. This extends a corresponding result of Thomas [69] in trace theory. Moreover, this equivalence and also a Kleene-type characterization of the recognizable languages even hold in the set $\mathrm{M}^{\infty}(\mathcal{A})$ of all finite and infinite computations. This extends the classical result of Büchi [11, 10] for infinite words and the results of Ebinger and Muscholl [32] and Gastin, Petit and Zielonka [34] for infinite traces.

Guaiana, Restivo and Salemi [38] extended Schützenberger's well-known result on the coincidence of the aperiodic and starfree word languages to trace monoids. In concurrency monoids, this coincidence fails in general even if $\mathcal{A}$ is assumed to be stably concurrent. However, it can be derived under a suitable additional assumption on $\mathcal{A}$ [22]. Also, we can show that aperiodic languages in large classes (but not all) of concurrency monoids are first-order definable, and vice versa [24]. This shows a borderline how far McNaughton and Papert's result [55] on such word languages and the trace-theoretic result of Thomas [69] and Ebinger and Muscholl [32] can be extended to concurrency monoids. We note that whereas in trace monoids these classes of languages coincide, in concurrency monoids the specific assumptions on the underlying automaton $\mathcal{A}$ become important.

We note that also a temporal logic for concurrent computations of stably concurrent automata can be naturally defined. Finally, we point out that for a suitable class of stably concurrent automata, somewhat surprisingly the satisfiability problems becomes elementary. This is in sharp contrast to the situation for traces [70] . Hence these stably concurrent automata might provide models for concurrent systems with (at least theoretically) much better tractable verification procedures for formulae of temporal logics than given in trace theory.

2 Definitions, the residuum operation and Petri nets

In the study of concurrent processes labeled transition systems have been used frequently as a model for an operational semantics [39, 56, 72]. A labeled transition system may be defined to be a triple $\mathcal{T} = (Q, \Sigma, T)$ where Q is a set of states, Σ is a set of actions and $T \subseteq Q \times \Sigma \times Q$ is a set of transitions. Several attempts have been made to incorporate direct information about concurrency into such a model. Stark [67] considered a symmetric irreflexive binary relation $\|$ on the actions to describe when two actions can be executed concurrently. A deterministic labeled transition system with such a relation is called trace automaton. Here, we generalize this model into two directions. First, we consider nondeterministic labeled transition systems. And secondly, we replace the single global concurrency relation by a family of concurrency relations, indexed by the states of the system. In this new model, it is possible that two actions occur independently in one state, but not in another one.

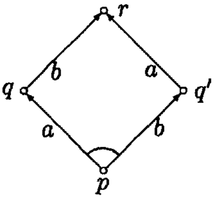
The formal definition of our model is as follows:

Definition 2.1 A *nondeterministic automaton with concurrency relations* is a quadruple $\mathcal{A} = (Q, \Sigma, T, \|)$ where

1. Q and Σ are countable disjoint nonempty sets of *states* (or situations) and of *actions*, respectively.
2. $T \subseteq Q \times \Sigma \times Q$ is a set of transitions.
3. $\| = (\|_q)_{q \in Q}$ is a family of irreflexive, symmetric binary relations on Σ such that whenever $a \|_p b$ there exist states $q, q', r \in Q$ and transitions (p, a, q), (p, b, q'), (q, b, r), (q', a, r) in T.

A nondeterministic automaton with concurrency relations $\mathcal{A} = (Q, \Sigma, T, \|)$ is *deterministic* if $(p, a, q), (p, a, r) \in T$ imply $q = r$. In this case, we call $\mathcal{A}$ a *deterministic automaton with concurrency relations* or simply an *automaton with concurrency relations*.

A transition $t = (p, a, q) \in T$ intuitively represents a potential computation step in which action a is executed in state p of $\mathcal{A}$ and $\mathcal{A}$ changes to state q. The *concurrency relations* $\|_q$ describe the concurrency information for pairs of actions at state q. The last requirement can be seen as in the diagram:



The angle at p indicates that $a \|_p b$ holds. A computation sequence is either empty (denoted by ε) or a finite or infinite sequence of transitions $u = t_1 t_2 \dots t_n$ or $u = t_1 t_2 \dots$ of the form $t_i = (q_{i-1}, a_i, q_i)$ for $i = 1, 2, \dots, n$ ($i \in \mathbb{N}$, respectively). The state q_0 is called *domain* of u (denoted by $\operatorname{dom} u$), and $a_1 a_2 \dots$ is the *action sequence* of u (denoted $\operatorname{acseq} u$). If u is finite, the state q_n is the *codomain* of u (denoted $\operatorname{cod} u$) and n is the length of u (denoted $|u|$). For infinite u, we write $|u| = \omega$. We let $\mathrm{CS}(\mathcal{A})$ denote the set of all finite computation sequences, $\mathrm{CS}^\omega(\mathcal{A})$ denotes the set of all infinite computation sequences and $\mathrm{CS}^\infty(\mathcal{A}) = \mathrm{CS}(\mathcal{A}) \cup \mathrm{CS}^\omega(\mathcal{A})$. Let $u \in \mathrm{CS}(\mathcal{A})$ and $v \in \mathrm{CS}^\infty(\mathcal{A})$. Then, the *composition* uv is defined in the natural way by concatenating u and v if $\operatorname{cod} u = \operatorname{dom} v$. Formally, we put $u\varepsilon = u$ and $\varepsilon v = v$. A finite computation sequence u is a *prefix* of the computation sequence w if there exists $v \in \mathrm{CS}^\infty(\mathcal{A})$ such that $w = uv$.

Now we want the concurrency relations of $\mathcal{A}$ to induce an equivalence relation on $\mathrm{CS}(\mathcal{A})$. This construction is similar to that of traces. Let $\sim$ denote the congruence with respect to concatenation on $\mathrm{CS}(\mathcal{A})$ generated by indentifying all computation sequences $(p,a,q)(q,b,r)$ and $(p,b,q')(q',a,r)$ with $a \parallel_p b$. Observe that $\sim$ is an equivalence relation on $\mathrm{CS}(\mathcal{A})$ such that two equivalent computation sequences have the same length, domain and codomain. We let $[u]$ denote the equivalence class of u with respect to $\sim$. Also, we let $1 := [\varepsilon]$. Defining $\mathrm{M}(\mathcal{A}) = \mathrm{CS}(\mathcal{A})/{\sim} \cup \{0\}$, we now obtain the *monoid* $\mathrm{M}(\mathcal{A})$ *of computations associated with* $\mathcal{A}$, where 0 is an additional symbol. That is, for $u, v \in \mathrm{CS}(\mathcal{A})$ we have $[u]\cdot[v] = [uv]$ if $\operatorname{cod} u = \operatorname{dom} v$ and $[u]\cdot[v] = 0$ otherwise. Also, $x \cdot 0 = 0 \cdot x = 0$ and $x \cdot 1 = 1 \cdot x = x$ for any $x \in \mathrm{M}(\mathcal{A})$. Clearly, with this operation $\mathrm{M}(\mathcal{A})$ is a monoid with 1 as unit (and with 0 as zero).

Next we show why these automata and their monoids provide a generalization of trace alphabets and trace monoids. Let (Σ, I) be a trace alphabet, *i.e.* Σ is a set and I an irreflexive symmetric relation on Σ. Let $\sim_I$ be the congruence on Σ^* generated by $\{(ab, ba) \mid (a,b) \in I\}$. Then the quotient $M(\Sigma, I) := \Sigma^*/{\sim_I}$ is called *trace monoid* over (Σ, I). Now $\mathcal{A} = (Q, \Sigma, T, \parallel)$ with $Q = \{q\}$, $T = Q \times \Sigma \times Q$ and $\parallel_q = I$ is an automaton with concurrency relations. This automaton will be called the *automaton induced by* (Σ, I). Obviously, $\mathrm{acseq} : \mathrm{CS}(\mathcal{A}) \to \Sigma^*$ is a bijection. Moreover, for $u, v \in \mathrm{CS}(\mathcal{A})$ we have $u \sim v$ iff $(\mathrm{acseq}\, u) \sim_I (\mathrm{acseq}\, v)$. Hence, acseq induces a monoid isomorphism from $\mathrm{M}(\mathcal{A}) \setminus \{0\}$ onto $M(\Sigma, I)$. In this sense, automata with concurrency relations generalize trace alphabets.

Now we want to extend the equivalence relation $\sim$ to an equivalence relation on $\mathrm{CS}^\infty(\mathcal{A})$. For $u, w \in \mathrm{CS}(\mathcal{A})$ let $u \lesssim w$ whenever there exists $v \in \mathrm{CS}^0(\mathcal{A})$ with $w \sim uv$. This relation is a preorder on $\mathrm{CS}(\mathcal{A})$ with $\sim \; = \; \lesssim \cap \gtrsim$. For $u, v \in \mathrm{CS}^\infty(\mathcal{A})$ we define $u \lesssim v$ if for any finite prefix u' of u there exists a finite prefix v' of v with $u' \lesssim v'$. The use of the same symbol is justified since for $u, v \in \mathrm{CS}(\mathcal{A})$ these two definitions yield the same relation. Moreover, $\lesssim$ is a preorder on $\mathrm{CS}^\infty(\mathcal{A})$. Hence $\sim \; := \; \lesssim \cap \gtrsim$ is an equivalence relation on $\mathrm{CS}^\infty(\mathcal{A})$. Again, the symbol $\sim$ is justified since for finite computation sequences this definition coincides with the original one as remarked above. Obviously, $u \sim v$ implies $\operatorname{dom} u = \operatorname{dom} v$, $|u| = |v|$ and (if $|u| < \omega$) $\operatorname{cod} u = \operatorname{cod} v$ for $u, v \in \mathrm{CS}^\infty(\mathcal{A})$.

Let $\mathrm{M}^\omega(\mathcal{A}) = \mathrm{CS}^\omega(\mathcal{A})/{\sim}$ and $\mathrm{M}^\infty(\mathcal{A}) = \mathrm{M}(\mathcal{A}) \cup \mathrm{M}^\omega(\mathcal{A})$. The elements of $\mathrm{M}^\infty(\mathcal{A})$ are called *computations*. The preorder $\lesssim$ on $\mathrm{CS}^\infty(\mathcal{A})$ induces a partial order on $\mathrm{M}^\infty(\mathcal{A})$ by letting $[u] \le [v]$ iff $u \lesssim v$, for any $u, v \in \mathrm{CS}^\infty(\mathcal{A})$.

Distinguishing an *initial state* $\star \in Q$ we obtain a nondeterministic automaton with concurrency relations and initial state $\mathcal{A} = (Q, \Sigma, T, \parallel, \star)$. In this case, $\mathrm{CS}_\star^\infty(\mathcal{A})$ denotes the set of all *initial computation sequences, i.e.* all $u \in \mathrm{CS}^\infty(\mathcal{A})$ with $u = \varepsilon$ or $\operatorname{dom} u = \star$. Moreover, $\mathrm{CS}_\star(\mathcal{A}) := \mathrm{CS}_\star^\infty(\mathcal{A}) \cap \mathrm{CS}(\mathcal{A})$. The quotient $\mathrm{CS}_\star^\infty(\mathcal{A})/{\sim}$ is denoted by $\mathrm{D}(\mathcal{A})$. The partial order $(\mathrm{D}(\mathcal{A}), \le)$ is the *domain generated by* $\mathcal{A}$. It is easy to see that $(\mathrm{D}(\mathcal{A}), \le)$ is indeed a domain, *i.e.* an ω-algebraic cpo.

Another possibility to incorporate direct information about concurrency was considered by Panangaden and Stark [61]. They endowed a deterministic labeled transition system with a residuum operation. This is a partial function $\uparrow$ mapping pairs of transitions to transitions and describes what remains of a transition s after performing a transition t. It can be extended to a corresponding mapping on pairs of finite computation sequences, and the properties of this mapping are crucial in the proofs of the basic results on $\mathrm{D}(\mathcal{A})$ and $\mathrm{M}(\mathcal{A})$.[1]

---

[1] In the context of group and monoid presentations, Dehornoy investigates a similar operation [14].

For a deterministic automaton with concurrency relations $\mathcal{A}$, we can define a similar residuum operation: First, the residuum of s and s equals ε, the empty computation sequence, for any transition s. The residuum of two different transitions (p, a, q) and (p', b, q') is defined iff $p = p'$ (*i.e.* they have the same start state) and $a \parallel_p b$. In this case, as in the picture following Def. 2.1, there exist uniquely determined transitions (q', a, r) and (q, b, r) which we take as $(p, a, q) \uparrow (p, b, q')$ and $(p, b, q') \uparrow (p, a, q)$, respectively. Furthermore $(p, a, q)(q, b, r) \sim (p, b, q')(q', a, r)$. It is easy to see that $s' = s \uparrow t$ iff there exists a transition t' such that $st' \sim ts'$ for any $s', s, t \in T$. Hence for nondeterministic automata, it is natural to construct a nondeterministic version of the residuum operation, *i.e.* a binary operation on T taking values in the powerset of $T \cup \{\varepsilon\}$, as follows:

$$s - t = \begin{cases} \{s' \in T \mid \exists t' \in T : st' \sim ts'\} & \text{if } s \neq t \\ \{\varepsilon\} & \text{otherwise.} \end{cases}$$

It can be checked that in a deterministic automaton with concurrency relations, $s - t = \{s \uparrow t\}$ if $s \uparrow t$ is defined, and $s - t = \emptyset$ otherwise. This residuum is in general no residuum operation in the sense of [61, 66] except if $\mathcal{A}$ satisfies the cube axiom. In terms of this residuum operation $-$, we can define nondeterministic concurrent automata:

Definition 2.2 A nondeterministic automaton with concurrency relations $\mathcal{A}$ satisfies the *cube axiom* or is a *nondeterministic concurrent automaton* if

$$tu \sim t'u' \implies (s - t) - u = (s - t') - u'$$

for any transitions $s, t, u, t', u' \in T$ (here, we let $\varepsilon - s = \{\varepsilon\}$ for any $s \in T$).

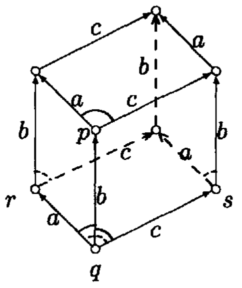
We remark that a deterministic automaton with concurrency relations $\mathcal{A}$ satisfies the cube axiom as just defined iff it satisfies the following implication:

$$a \parallel_q b,\ b \parallel_q c \text{ and } a \parallel_{q.b} c \implies a \parallel_q c,\ b \parallel_{q.a} c \text{ and } a \parallel_{q.c} b. \qquad (*)$$

(where $q.b$ is the unique state satisfying $(q, b, q.b) \in T$ – in the same spirit, we write $q.w$ for the unique state reached from q by executing $w \in \Sigma^*$ if such a state exists.)

The proof of this is simple but tedious. It rests on the definition of the residuum operation through the concurrency relations described above.

Suppose the assumptions of the implication above hold. Then, by the requirement 3 for a nondeterministic automaton with concurrency relations, the transitions marked by solid lines in the following picture exist. The conclusion together with requirement 3 implies the existence of the transitions depicted by dotted lines exist. This picture suggests the name "cube axiom".



It turns out that for nondeterministic concurrent automata, we can extend the residuum operation to one on $\mathrm{CS}(\mathcal{A})$ such that it respects equivalence, *i.e.* $u, v, u', v' \in \mathrm{CS}(\mathcal{A})$ and $u \sim u'$, $v \sim v'$ imply that $u - v \sim u' - v'$ (meaning that for each $w \in u - v$ there is $w' \in u' - v'$ with $w \sim w'$ and vice versa), see [44, pp. 122-135]. This makes algebraic calculations with the elements of $\mathrm{CS}(\mathcal{A})$ and $\mathrm{M}(\mathcal{A})$ possible, which is crucial *e.g.* for the proof of Thm. 3.1 below (see also [14] for a similar, but more general, calculus in semigroups).

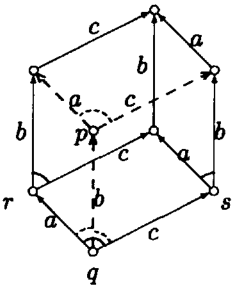
Next we consider an axiom which for deterministic automata is precisely the converse of condition $(*)$.

Definition 2.3 A nondeterministic automaton with concurrency relations $\mathcal{A}$ satisfies the *inverse cube axiom* if $t_1u_1 \sim t_2u_2$, $t_1 \neq t_2$ and $(s_1 - u_1) \cap (s_2 - u_2) \neq \emptyset$ imply the existence of $s \in T$ such that $s_1 \in s - t_1$ and $s_2 \in s - t_2$ for any $s_1, s_2, t_1, t_2, u_1, u_2 \in T$.

As indicated, for deterministic automata with concurrency relations there exists a simpler form of this axiom:

$$a \parallel_q b,\ b \parallel_q c \text{ and } a \parallel_{q.b} c \iff a \parallel_q c,\ b \parallel_{q.a} c \text{ and } a \parallel_{q.c} b \qquad (*^{-1})$$

for any $a, b, c \in \Sigma$ and $q \in Q$. This converse implication, as well as the following picture (where we make the same conventions as above on solid and dotted lines) may justify the name "inverse" cube axiom.



The importance of this axiom will be clear from Thm. 3.4. A similar axiom has been introduced by Panangaden, Shanbhogue and Stark in [60] for the more specific situation of input/output automata.

A (non-)deterministic automaton with concurrency relations that satisfies the cube and the inverse cube axiom is called *(non-)deterministic stably concurrent automaton.* Since any automaton induced by a trace alphabet has precisely one state, it is deterministic and satisfies the cube and the inverse cube axiom. Hence, deterministic stably concurrent automata still generalize trace alphabets.

For later purposes we now extend the concurrency relations $\parallel_q$ inductively to words over Σ. Let $\mathcal{A}$ be a deterministic automaton with concurrency relations. Let $a \in \Sigma$ and $v, w \in \Sigma^+$. Then $av \parallel_q w \iff a \parallel_q w$ and $v \parallel_{q.a} w$. Furthermore, $a \parallel_q w$ iff $w \parallel_q a$. For $u, v \in \mathrm{CS}(\mathcal{A})$ we say *u and v commute* (denoted $u \parallel v$) iff $\operatorname{dom} u = \operatorname{dom} v =: q$ and $(\operatorname{acseq} u) \parallel_q (\operatorname{acseq} v)$. One can show that then also v and u commute (*cf.* [21]).

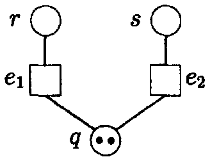
Nielsen, Rosenberg and Thiagarajan [58] established a close relationship between a class of transition systems and elementary Petri nets. In [72], Winskel and Nielsen established an adjunction between the category of Petri nets (essentially, condition/event nets) and the category of asynchronous transition systems. For a particular subclass of asynchronous transition systems, this adjunction even turned out to be a coreflection. Mukund [57] considered nets with capacities (place/transition nets) in which events (transitions) can fire with multiplicities and places have unbounded capacities, and obtained for a suitable class of transition systems a similar coreflection. In the rest of this section, we present a result on the relationship of deterministic automata with concurrency relations and place/transition nets.

A *place/transition net* or, for short, a *PT-net* is a quintuple $\mathcal{N} = (P, \Sigma, F, M_0)$ such that

1. P and Σ are disjoint sets of *places* and *actions*, respectively.
2. $F : (P \times \Sigma) \cup (\Sigma \times P) \to \mathbb{N}$ is a function.
3. $M_0 : P \to \mathbb{N}$ is the initial marking of the net $\mathcal{N}$.

A marking of $\mathcal{N}$ is a function $M : P \to \mathbb{N}$.

Consider the following net. We assume $F(q, e_1) = F(q, e_2) = F(e_1, r) = F(e_2, s) = 1$. All other values of F are zero. Suppose, M is a marking of the net with $M(q) = 2$. Then both actions e_1 and e_2 can consume one token from q, *i.e.* they can fire simultanously. If $M(q) = 1$, only one of these actions can take place. Therefore, possible concurrency of actions is state-dependent in a PT-net.



Let us indicate how to construct an automaton with concurrency relations $\mathcal{A} = na(\mathcal{N})$ from a net $\mathcal{N} = (P, \Sigma, F, M_0)$. Let Q comprise all markings of the net $\mathcal{N}$. The marking M_0 is the initial state $\star$ of $\mathcal{A}$. Let $M, \overline{M} \in Q$ and $e_1 \in \Sigma$. Then $(M, e_1, \overline{M})$ is a transition iff $M(p) \geq F(p, e_1)$ and

$$\overline{M}(p) = M(p) - F(p, e_1) + F(e_1, p)$$

for any $p \in P$. Furthermore, for $e_1, e_2 \in \Sigma$ with $e_1 \neq e_2$ and $M \in Q$ we have $e_1 \parallel_M e_2$ iff $M(p) \geq F(p, e_1) + F(p, e_2)$ for any $p \in P$ and the markings $M.e_1$, $M.e_2$ and $M.e_1e_2 = M.e_2e_1$ exist. This intuitively says that there are enough recources such that e_1 and e_2 can fire simultanously in the net $\mathcal{N}$.

Therefore automata with concurrency relations and their computation sequences seem to be a natural model for the dynamic behaviour of PT-nets, similarly as traces model the behaviour of condition/event-nets.

PT-nets are an intensively studied model of concurrent processes, *cf.* [63]. The class of PT-nets can be endowed with morphisms as follows: a pair (α, β) is a *net-morphism* from the PT-net $\mathcal{N}$ to the PT-net $\mathcal{N}'$ whenever $\alpha : P' \to P$ and $\beta : \Sigma \to \Sigma'$ are functions such that $M_0 \circ \alpha = M_0'$, $F'(\beta(e), p') = F(e, \alpha(p'))$ and $F'(p', \beta(e)) = F(\alpha(p'), e)$ for any $p' \in P'$ and $e \in \Sigma$. This defines a category $\mathbb{P}$.

Between deterministic automata with concurrency relations and initial states $\mathcal{A}$ and $\mathcal{A}'$ we can define morphisms applying the usual constructions from model theory or universal algebra: let $\pi : Q \to Q'$ and $\eta : \Sigma \to \Sigma'$ be functions such that

1. $(p, a, q) \in T$ implies $(\pi(p), \eta(a), \pi(q)) \in T'$.

2. $\pi(\star) = \star'$.

3. $a \parallel_p b$ in $\mathcal{A}$ implies $\eta(a) \parallel'_{\pi(p)} \eta(b)$ in $\mathcal{A}'$.

The pair (π, η) is called ***automaton morphism.***

This concept of morphism defines a category $\mathbf{A}$. The two categories $\mathbb{P}$ and $\mathbf{A}$ are related as follows:

Theorem 2.4 **([30])** *There exist functors an* : $\mathbf{A} \to \mathbb{P}$ *and na* : $\mathbb{P} \to \mathbf{A}$ *that form an adjunction where an is the left adjoint of na.*

In [30], the authors describe a full subcategory $\mathbf{A}_0$ of $\mathbf{A}$ and a functor $na_0 : \mathbb{P} \to \mathbf{A}_0$ such that $an \restriction \mathbf{A}_0$ and na_0 form a coreflection.

Moreover, in [28], similar results were established for nets with capacities and markings which could even take values in partially ordered abelian groups. In [29] a continuous version of P/T-nets and of continuous transition systems with situation-dependent concurrency is investigated.

3 Domains

For a PT-net $\mathcal{N}$, usually the set of all initial firing sequences is considered as the language of the net. Similarly, the set of initial computations $\mathrm{D}(\mathcal{A})$ can be regarded as the "language" of a nondeterministic automaton with concurrency relations $\mathcal{A}$. This set carries a nontrivial partial order $\leq$ as defined in Sect. 2.

In this section, we summarize the results on $(\mathrm{D}(\mathcal{A}), \leq)$ for nondeterministic automata with concurrency relations. Therefore, we first introduce some order-theoretic notations.

Let $(D, \leq)$ be a partially ordered set. Let $x, y \in D$. Then y *covers* x (denoted $x {-\!\!<} y$) if $x < y$ and $x < z \leq y$ implies $y = z$. We put $x{\downarrow} := \{d \in D \mid d \leq x\}$. A subset $A \subseteq D$ is *directed* if for any $x, y \in A$ there is $z \in A$ such that $x, y \leq z$. $(D, \leq)$ is a *complete partial order* or cpo, if it has a least element (usually denoted by $\perp$) and any directed subset of D has a supremum. An element of a cpo $(D, \leq)$ is *compact* if any directed subset A of D with $\sup A \geq x$ contains some $a \in A$ with $a \geq x$. The set D^0 comprises all compact elements of D. The cpo $(D, \leq)$ is *algebraic* if for any $x \in D$, the set $X = x{\downarrow} \cap D^0$ is directed and $\sup X = x$. A *domain* is an algebraic cpo with at most countably many compact elements. A domain $(D, \leq)$ is *finitary* if $x{\downarrow}$ is finite for any $x \in D^0$.

A partially ordered set $(D, \leq)$ is a *lattice* if any finite subset has a supremum and an infimum. It *has finite height* if any totally ordered subset of D is finite. A lattice of finite height is *semimodular* if whenever $x, y_1, y_2 \in D$ with $x {-\!\!<} y_1, y_2$ and $y_1 \neq y_2$, then $y_1 {-\!\!<} y_1 \vee y_2$. (In domain theory, semimodularity is known as axiom (C).) It is *modular* if moreover $y_1 \wedge y_2 {-\!\!<} y_1$ for any $y_1, y_2, z \in D$ with $y_1, y_2 {-\!\!<} z$ and $y_1 \neq y_2$. A lattice is *distributive* if $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ for any $x, y, z \in D$. Recall that any distributive lattice of finite height is modular and finite.

Let $\mathcal{A}$ be a nondeterministic automaton with concurrency relations and initial state. Then $(\mathrm{D}(\mathcal{A}), \leq)$ is a domain. Its compact elements are the finite initial computations, *i.e.* $\mathrm{D}(\mathcal{A})^0 = \mathrm{D}(\mathcal{A}) \cap \mathrm{M}(\mathcal{A})$. Hence, any compact element dominates compact elements only. If any set of the form $\{q \in Q \mid (p, a, q) \in T\}$ for $p \in Q$ and $a \in \Sigma$ is finite, $(\mathrm{D}(\mathcal{A}), \leq$ $)$ is moreover finitary. Obviously, this is the case for deterministic automata. In [19, 20] and [44] complete characterizations of the domains $(\mathrm{D}(\mathcal{A}), \leq)$ generated by (non-)

deterministic automata with concurrency relations and initial state are given. Here we discuss the impact of the cube axiom, the inverse cube axiom and the determinism of $\mathcal{A}$ to $(\mathrm{D}(\mathcal{A}), \leq)$.

Theorem 3.1 ([44]) *Let $\mathcal{A}$ be a nondeterministic concurrent automaton with initial state. Then, for any $x \in \mathrm{D}(\mathcal{A})^0$, $(x\!\downarrow, \leq)$ is a semimodular lattice. Furthermore, $(x\!\downarrow, \leq)$ is modular for any $x \in \mathrm{D}(\mathcal{A})^0$ iff $\mathcal{A}$ is stably concurrent.*

Conversely, let $(D, \leq)$ be a domain where any compact element dominates a finite distributive lattice. Then there exists a nondeterministic stably concurrent automaton with initial state $\mathcal{A}$ such that $(\mathrm{D}(\mathcal{A}), \leq) \cong (D, \leq)$.

Domains where any compact element dominates a lattice are known as L-domains. With continuous functions as morphisms they form an important category in domain theory (*cf.* [41]).

As noted before, the residuum operation of computations defined in Sect. 2 is an essential tool in the proof of the theorem above. This residuum operation can also be used to show the following cancellation properties of the monoid $\mathrm{M}(\mathcal{A})$:

Theorem 3.2 ([67, 19, 9]) *Let $\mathcal{A}$ be a deterministic concurrent automaton. Then $\mathrm{M}(\mathcal{A})$ is a cancellative monoid.*

For nondeterministic concurrent automata $\mathcal{A}$, in general $\mathrm{M}(\mathcal{A})$ is just left-cancellative, but not necessarily right-cancellative, as examples show. Right-cancellation holds whenever $\mathcal{A}$ is stably concurrent.

Now we turn to deterministic automata with concurrency relations. A domain $(D, \leq)$ is a Scott-domain if any two elements bounded above have a supremum. Let $(D, \leq)$ be a domain and $x, y \in D^0$. If $x \longrightarrow\!\!< y$, we call $[x, y]$ a *prime interval*. Let $>\!\!\longrightarrow\!\!<$ denote the smallest equivalence relation on the set of all prime intervals such that $[x, y] >\!\!\longrightarrow\!\!< [x', y']$ for all prime intervals $[x, y]$ and $[x', y']$ with $x \longrightarrow\!\!< x'$, $y \longrightarrow\!\!< y'$ and $y \neq x'$. A domain $(D, \leq)$ satisfies axiom (R) if $[x, y] >\!\!\longrightarrow\!\!< [x, z]$ implies $y = z$ for any $x, y, z \in D$. A finitary Scott-domain satisfies axiom (C) if the lattice $(x\!\downarrow, \leq)$ for any compact element x is semimodular. A *concurrency domain* is a finitary Scott-domain $(D, \leq)$ satisfying axioms (R) and (C).

Theorem 3.3 ([19, 20]) *Let $(D, \leq)$ be a partially ordered set. Then $(D, \leq)$ is a concurrency domain iff there exists a deterministic concurrent automaton with initial state $\mathcal{A}$ such that $(D, \leq) \cong (\mathrm{D}(\mathcal{A}), \leq)$.*

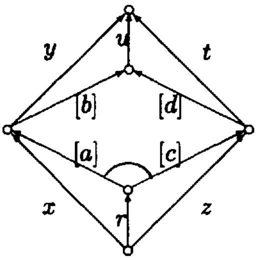
It has been shown that the domain of configurations of an event structure is a concurrency domain (*cf.* [71]). Hence for any event structure there exists a deterministic concurrent automaton with initial state $\mathcal{A}$ that generates an isomorphic domain (but not vice versa). In this sense, deterministic concurrent automata are more general than event structures. An event structure that generates an infinite domain is always infinite. On the other hand, an automaton with concurrency relations and initial state generates an infinite domain as soon as it has a loop. Hence the representation of domains by automata is much more compact than that by event structures.

In trace theory, Levi's Lemma plays a central role. In the theory of deterministic concurrent automata, it can be reformulated as follows:

A deterministic concurrent automaton $\mathcal{A}$ satisfies *Levi's Lemma* if for any $x, y, z, t \in \mathrm{M}(\mathcal{A})$ with $xy = zt \neq 0$ there exist uniquely determined elements $r, [a], [b], [c], [d], u \in$

M($\mathcal{A}$) such that $x = r[a]$, $y = [b]u$, $z = r[c]$, $t = [d]u$, $a \parallel c$, $\mathrm{acseq}\, a = \mathrm{acseq}\, d$ and $\mathrm{acseq}\, c = \mathrm{acseq}\, b$.

This definition can be depicted by the following diagram:



The following theorem establishes the close relation between Levi's Lemma, the distributivity of $(\mathrm{D}(\mathcal{A}), \leq)$ and, as a direct consequence of Thm. 3.1, the inverse cube axiom. A finitary Scott-domain where any element dominates a distributive lattice is a *dI-domain*.

Theorem 3.4 **([44, 21])** *Let $\mathcal{A}$ be a deterministic concurrent automaton with initial state. Then the following are equivalent:*

1. *$\mathcal{A}$ is stably concurrent.*
2. *$(\mathrm{D}(\mathcal{A}), \leq)$ is a dI-domain.*
3. *$\mathcal{A}$ satisfies Levi's Lemma.*

Together with Thm. 3.3, this theorem implies that the class of all domains generated by deterministic stably concurrent automata with initial states and the class of all dI-domains coincide. Hence we obtained a new representation result for the well studied class of dI-domains (*cf.* [13]). This result generalizes *e.g.* [18, Thm. 11.3.11]. Note that there are dI-domains that cannot be generated by a trace alphabet: in [5], a universal dI-domain is constructed that arises from a trace alphabet. While there exists a universal homogeneous dI-domain (which can, as any dI-domain, be generated by a stably concurrent automaton), it cannot be generated by a trace alphabet [46].

Next we consider the class of all deterministic automata with concurrency relations and initial state that generate up to isomorphism a given domain. This class may contain finite and infinite automata. The question is if we can find minimal automata in this class and if they are unique.

Let $\mathcal{A}_i = (Q_i, \Sigma, T_i, \parallel^i, \star_i)$ for $i = 1, 2$ be deterministic automata with concurrency relations and initial state; note that $\mathcal{A}_1$ and $\mathcal{A}_2$ are required here to have the same set of actions Σ. A surjective function $f : Q_1 \to Q_2$ is a *reduction* if

1. $f(\star_1) = \star_2$.
2. $\forall (p, a, q) \in T_1 : (f(p), a, f(q)) \in T_2$.
3. $\forall a, b \in \Sigma\ \forall q \in Q_1 : a \parallel^1_q b$ in $\mathcal{A}_1 \iff a \parallel^2_{f(q)} b$ in $\mathcal{A}_2$.
4. $\forall p \in Q_1\ \forall (f(p), a, q') \in T_2\ \exists q \in Q_1 : (p, a, q) \in T_1$.

In particular, (f, id_Σ) is a morphism from $\mathcal{A}_1$ to $\mathcal{A}_2$ as defined before. By requirement 2, a reduction $f : \mathcal{A}_1 \to \mathcal{A}_2$ determines a function $\overline{f} : \mathrm{CS}^\infty(\mathcal{A}_1) \to \mathrm{CS}^\infty(\mathcal{A}_2)$. By requirement 3, equivalent computation sequences of $\mathcal{A}_1$ are mapped to equivalent computation sequences of $\mathcal{A}_2$. Therefore, $\overline{f}$ can be considered as a function from $\mathrm{M}^\infty(\mathcal{A}_1)$ to $\mathrm{M}^\infty(\mathcal{A}_2)$. This mapping determines an isomorphism between the generated domains. Therefore, $(\mathrm{D}(\mathcal{A}_1), \leq) \cong (\mathrm{D}(\mathcal{A}_2), \leq)$.

Since a reduction $f : \mathcal{A}_1 \to \mathcal{A}_2$ is surjective, $\mathcal{A}_1$ has more states then $\mathcal{A}_2$, *i.e.* f decreases the number of states. Therefore, we can consider $\mathcal{A}_2$ as "smaller" than $\mathcal{A}_1$. Consequently, $\mathcal{A}_1$ is *minimal* if any reduction $f : \mathcal{A}_1 \to \mathcal{A}_2$ is injective, *i.e.* $\mathcal{A}_1$ has a minimal set of states.

Theorem 3.5 ([8]) *Let $\mathcal{A}$ be a deterministic automaton with concurrency relations and initial state. Then there exists, up to isomorphism, a uniquely determined minimal deterministic automaton with concurrency relations and initial state $\mathcal{A}'$ such that $\mathcal{A}$ can be reduced to $\mathcal{A}'$.*

The unique existence of the minimal automaton $\mathcal{A}'$ has been shown in [8] for a slightly restricted class of automata. In [45], an algorithm is presented that computes this minimal automaton from $\mathcal{A}$ (if $\mathcal{A}$ is finite). Actually, these papers deal with more general reductions that also decrease the set of actions. [45] proves similar results for nondeterministic automata.

We finish this section considering subautomata. Let $\mathcal{A}_i = (Q_i, \Sigma_i, T_i, \|^i, \star_i)$ for $i = 1, 2$ be deterministic concurrent automata with initial states. Then $\mathcal{A}_1$ is a *subautomaton* of $\mathcal{A}_2$ (denoted $\mathcal{A}_1 \subseteq \mathcal{A}_2$) if

1. $\Sigma_1 \subseteq \Sigma_2$, $Q_1 \subseteq Q_2$, and $\star_1 = \star_2$.
2. $(p, a, q) \in T_1 \iff (p, a, q) \in T_2$ for any $p \in Q_1$, $a \in \Sigma_1$ and $q \in Q_2$ (*i.e.* $T_1 = T_2 \cap (Q_1 \times \Sigma_1 \times Q_2)$).
3. $a \|_q^1 b$ in $\mathcal{A}_1 \iff a \|_q^2 b$ in $\mathcal{A}_2$ for any $a, b \in \Sigma_1$ and $q \in Q_1$.

We may consider automata as functional structures where the actions operate as partial functions on the set of states. Then $\mathcal{A}_1 \subseteq \mathcal{A}_2$ should imply that Q_1 is closed under the application of elements of Σ_1 in Q_2. This is satisfied since requirement 2 especially implies that $p.a$ in $\mathcal{A}_1$ is defined iff it is defined in $\mathcal{A}_2$ (for $p \in Q_1$ and $a \in \Sigma_1$).

To descibe the relation between the domains generated by $\mathcal{A}_1$ and $\mathcal{A}_2$, we need another order-theoretic notation:

Let $(D, \leq)$ be a concurrency domain. A subset S of D is a *nice ideal* if

1. $\forall A \subseteq S\ \forall d \in D : A \leq d \Rightarrow \exists s \in S : A \leq s \leq d.$
2. $\forall s \in S\ \forall d \in D : d \leq s \Rightarrow d \in S.$
3. $\forall x, x', y \in D^0 \cap S\ \forall y' \in D^0 : [x, x'] \rightarrowtail\!\!\!\prec [y, y'] \Rightarrow y' \in S.$

Theorem 3.6 ([20]) *Let $\mathcal{A}_i = (Q_i, \Sigma_i, T_i, \|^i, \star_i)$ for $i = 1, 2$ be deterministic concurrent automata with initial states such that $\mathcal{A}_1 \subseteq \mathcal{A}_2$. Then $\mathrm{D}(\mathcal{A}_1)$ is a nice ideal in $(\mathrm{D}(\mathcal{A}_2), \leq)$.*

Let $(D, \leq)$ be a concurrency domain and S be a nice ideal in $(D, \leq)$. Then there exist deterministic concurrent automata with initial states $\mathcal{A}_1 \subseteq \mathcal{A}_2$ such that $(D, \leq) \cong (\mathrm{D}(\mathcal{A}_2), \leq)$ and $(S, \leq) \cong (\mathrm{D}(\mathcal{A}_1), \leq)$.

Domains and traces have also been considered from a topological point of view. Historically the first topology for traces was introduced by Kwiatkowska [51] who generalized the prefix metric of words. While the infinite traces are the metric completion of the finite ones, the drawback of this topology is that the multiplication is not continuous. This observation led Diekert to the introduction of complex and α-complex traces [17]. In [50], we extend his investigation of α-complex traces to automata with concurrency relations. Our main result is the construction of a metric on finite computations that makes the concatenation uniformly continuous. We also describe the elements of the metric completion as pairs consisting of a (possibly infinite) computation and a finite set of transitions of the underlying automaton.

4 Dependence orders

From now on, we consider only deterministic automata with concurrency relations. To simplify notation, they are called automata with concurrency relations, as mentioned before.

One of the foundational results of trace theory [54], used in many difficult applications, is that each element of the (algebraically defined) trace monoid has a graph-theoretical representation. Moreover, infinite traces are usually defined in terms of dependence graphs, which is in contrast (but equivalent) to our approach to define them as equivalence classes of infinite computation sequences. We will show that these representation results can be generalized to monoids associated with stably concurrent automata. **Therefore, for all of this section, let $\mathcal{A}$ be a fixed stably concurrent automaton.**

We first define a labeled partial order $\mathrm{DO}(u)$ for any $u \in \mathrm{CS}^{\infty}(\mathcal{A})$. Let $u \in \mathrm{CS}^{\infty}(\mathcal{A}) \setminus \{\varepsilon\}$. Analogously to trace theory we define the dependence order on those events that appear in u. This order should reflect when an earlier event has to appear before a later one, *i.e.* the earlier event is a necessary condition for the later one. Since an action a can appear several times in u we have to distinguish between the first, the second ... appearance of a. For a finite or infinite word w over Σ and $a \in \Sigma$ let $|w|_a$ denote the number of a's in w. Then define $|u|_a := |\operatorname{acseq} u|_a$ We abbreviate $a^i = (a, i)$ for $a \in \Sigma$ and $i \in \mathbb{N}$. The precise definition of the dependence order of u can now be given as follows. Let $\mathrm{Q}^u_a = \{a^i \mid i \in \mathbb{N}, 1 \leq i \leq |u|_a\}$ for $a \in \Sigma$ and $\mathrm{O}(u) = \bigcup\{\mathrm{Q}^u_a \mid a \in \Sigma\}$. Then, obviously, $|\mathrm{O}(u)| = |u|$. For $a^i, b^j \in \mathrm{O}(u)$ let $a^i \sqsubseteq_u b^j$ iff the i-th appearance of a in u precedes the j-th appearance of b, *i.e.*, formally, there are words $x, y \in \Sigma^*$ and a possibly infinite word z over Σ with $\operatorname{acseq} u = xaybz$, $|x|_a = i - 1$ and $|xay|_b = j - 1$. Then $\sqsubseteq_u$ is a linear order on $\mathrm{O}(u)$. Since for equivalent computation sequences u and v we always have $\mathrm{O}(u) = \mathrm{O}(v)$, a partial order on $\mathrm{O}(u)$ can be defined by:

$$\preceq_u := \bigcap\{\sqsubseteq_v \mid v \sim u\}.$$

Hence, $a^i \preceq_u b^j$ if and only if the i-th a precedes the j-th b in *any* computation sequence equivalent with u. Now $\mathrm{DO}(u) = (\mathrm{O}(u), \preceq_u, (\mathrm{Q}^u_a)_{a \in \Sigma}, \operatorname{dom} u)$ is a relational structure with one constant from Q. We call $\mathrm{DO}(u)$ the ***dependence order associated with u***. Since $u \sim v$ implies $\mathrm{DO}(u) = \mathrm{DO}(v)$, the dependence order $\mathrm{DO}(u)$ can be considered as the dependence order of the computation $[u] \in \mathrm{M}^{\infty}(\mathcal{A}) \setminus \{0, 1\}$. To include 0 and 1, formally we put $\operatorname{dom} 0 = \bot$ and $\operatorname{dom} 1 = \top$ where $\bot$ and $\top$ are additional symbols not belonging to Q, and, using this, define $\mathrm{DO}(0)$ and $\mathrm{DO}(1)$ similarly as before (with $\mathrm{O}(0) = \mathrm{O}(1) = \emptyset$).

A second labeled partial order $\mathrm{PR}(u)$ can be extracted from the distributive lattice $([u]\!\downarrow, \leq)$. A finite computation $x \in \mathrm{M}(\mathcal{A})$ is a *complete prime* if there exists precisely one finite computation y and a transition t such that $x = y \cdot [t]$. Let $\mathrm{Pr}(u)$ comprise all complete primes $x \leq [u]$. We only note that these are precisely the join-irreducible elements of the lattice $([u]\!\downarrow, \leq)$. By a fundamental result in lattice theory [4], they completely determine the structure of this lattice since it is distributive. For $a \in \Sigma$ let P^u_a comprise all elements x of $\mathrm{Pr}(u)$ such that $x = y \cdot [t]$ for some transition $t = (p, a, q)$ and some computation y. Furthermore, define $\mathrm{PR}(u) = (\mathrm{Pr}(u), \leq, (\mathrm{P}^u_a)_{a\in\Sigma}, \mathrm{dom}\, u)$. A foundational result on dependence orders is that, although defined quite differently, the labeled partial orders $\mathrm{DO}(u)$ and $\mathrm{PR}(u)$ are isomorphic for $u \in \mathrm{CS}^\infty(\mathcal{A})$. This isomorphism is also used to prove the following result on order preserving enumerations of $\mathrm{DO}(u)$: a (possibly infinite) sequence $A = (x_i)_{i<n}$ for $n \in \mathbb{N} \cup \{\omega\}$ is an *order-preserving enumeration* of $\mathrm{DO}(u)$ if it is an enumeration of $\mathrm{O}(u)$ and $x_i \preceq_u x_j$ implies $i \leq j$. Then a computation sequence v is the *linearisation of* $\mathrm{DO}(u)$ *induced by* A if $\mathrm{dom}\, u = \mathrm{dom}\, v$ and $\mathrm{acseq}\, v = a_1 a_2 \ldots a_{n-1}$ ($\mathrm{acseq}\, v = a_1 a_2 \ldots$, respectively) with $x_i \in \mathrm{Q}^u_{a_i}$, for $i < n$. We call v a *linearisation of* $\mathrm{DO}(u)$ if it is the linearisation induced by some order-preserving enumeration. Since $\mathcal{A}$ is deterministic, any order-preserving enumeration induces at most one linearisation. Let $\mathrm{Lin}\,\mathrm{DO}(u)$ comprise all linearisations of $\mathrm{DO}(u)$. Then it is easy to see that $[u] \subseteq \mathrm{Lin}\,\mathrm{DO}(u)$ for any $u \in \mathrm{CS}^\infty(\mathcal{A})$. (Actually, this holds even for arbitrary automata with concurrency relations $\mathcal{A}$.) If $\mathcal{A}$ is stably concurrent, we even have

Theorem 4.1 ([**9, 26**]) *Let $\mathcal{A}$ be a stably concurrent automaton and let $u, v \in \mathrm{CS}^\infty(\mathcal{A})$. Then $\mathrm{DO}(u) \cong \mathrm{PR}(u)$ and the following are equivalent.*

1. $u \sim v$.
2. $\mathrm{DO}(u) = \mathrm{DO}(v)$.
3. $u \in \mathrm{Lin}\,\mathrm{DO}(v)$.

Furthermore, any order-preserving enumeration of $\mathrm{DO}(u)$ *induces a linearisation.*

Hence, in this case $[u] = \mathrm{Lin}\,\mathrm{DO}(u)$. The importance of such a description of linearizations of the partial order for concurrent programs is discussed, *e.g.* in [42, 43]. This result enables us to represent computation sequences by their dependence orders. In [9], all those labeled partial orders are characterized that are isomorphic to the dependence order $\mathrm{DO}(u)$ of a finite computation sequence u. Moreover, a multiplication on the set of (isomorphism classes of) finite dependence orders is defined and shown to yield a monoid isomorphic to the monoid $\mathrm{M}(\mathcal{A})$. This generalizes classical results of Mazurkiewicz ([54, 16]).

Since dependence orders are relational structures, we can define logical languages to reason on these dependence orders. The corresponding first-order language FO has variables $x, y, \ldots$ for elements of $\mathrm{O}(u)$. The atomic formulas are $x \leq y$, $\Sigma_a(x)$ for $a \in \Sigma$, and constants c_q for $q \in Q \cup \{\bot, \top\}$. Then formulas are built up from atomic formulas by the connectives $\neg$ and $\vee$ and the quantifier $\exists$. In the monadic second order language MSO, also set variables $X, Y, \ldots$, quantification of them and atomic formulas $X(x)$ are admitted. A sentence of MSO is a formula without free variables. The satisfaction relation $\mathrm{DO}(u) \models \varphi$ between dependence orders and sentences is defined as usually: $x \leq y$ is satisfied iff $x \preceq_u y$, $\Sigma_a(x)$ iff $x \in \mathrm{Q}^u_a$, c_q iff $\mathrm{dom}(u) = q$ and $X(x)$ iff $x \in X$. Now let φ be a sentence of MSO. Then $L^\infty(\varphi)$ denotes the set of all $[u] \in \mathrm{M}^\infty(\mathcal{A})$

such that $\mathrm{DO}(u) \models \varphi$. Furthermore, $L(\varphi) = L^\infty(\varphi) \cap \mathbb{M}(\mathcal{A})$. Since $u \sim v$ implies $\mathrm{DO}(u) = \mathrm{DO}(v)$, the set $L^\infty(\varphi)$ is well defined. The logical languages FO and MSO will be used in the following sections.

5 Recognizable languages in $\mathbb{M}(\mathcal{A})$ and $\mathbb{M}^\infty(\mathcal{A})$

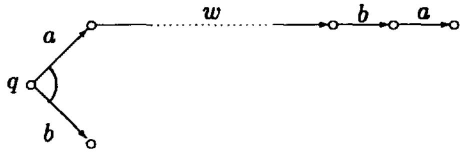
In Sect. 3, we considered the domain of initial computations of $\mathcal{A}$ as the "language" of the automaton $\mathcal{A}$. Here, we adopt another point of view: for an automaton with concurrency relations $\mathcal{A}$, $(\mathbb{M}(\mathcal{A}), \cdot)$ is a monoid. For a monoid there is a well studied concept of "recognizable language" (see below). In this section we describe the recognizable languages of the monoid $(\mathbb{M}(\mathcal{A}), \cdot)$, where $\mathcal{A}$ is a finite automaton with concurrency relations, by a calculus of certain rational expressions and by logical formulas.

Let $(M, \cdot)$ be a monoid. A language (subset) $L \subseteq M$ is *recognizable* if there exists a finite monoid $(S, \cdot)$ and a homomorphism $\eta : M \to S$ such that $\eta^{-1}\eta(L) = L$ [33]. Equivalently, there exists a finite M-automaton recognizing L.

In a classical result, Kleene characterized the recognizable languages in a finitely generated free monoid as precisely the rational languages. This was generalized to trace monoids by Ochmański [59]; here one has to replace the usual star-iteration by a concurrent iteration (which restricts to the usual iteration in the case of free monoids). Now we want to extend this result to the monoid $\mathbb{M}(\mathcal{A})$. It turns out that here the precise assumptions on $\mathcal{A}$ matter. For instance, if $\mathcal{A}$ is just assumed to be concurrent, the product of two recognizable languages in $\mathbb{M}(\mathcal{A})$ need not be recognizable (see [21] for an example).

We introduce some notation. Let $u, v \in \mathrm{CS}(\mathcal{A})$ with $\mathrm{dom}\, u = \mathrm{dom}\, v$. Then v is a *component* of u if there exists $w \in \mathrm{CS}(\mathcal{A})$ such that $v \parallel w$ and $[u] = [v] \vee [w]$ in $(\mathbb{M}(\mathcal{A}), \leq)$; this means that v, w commute and u is equivalent to an interleaving of v and w. The computation sequence u is *irreducible* if there are no non-trivial components of u. If $\mathcal{A}$ is stably concurrent, v is a component of u iff $\mathrm{dom}\, u = \mathrm{dom}\, v$ and $\mathrm{DO}(v)$ is a component of $\mathrm{DO}(u)$ (*i.e.* the elements of $\mathrm{O}(v)$ and $\mathrm{O}(u) \setminus \mathrm{O}(v)$ are pairwise incomparable). For $L \subseteq \mathbb{M}(\mathcal{A})$ let $\mathrm{IComp}(L)$ denote the set of all irreducible components v of elements u of L with $\mathrm{dom}\, v = \mathrm{cod}\, v$. Then the *concurrent iteration* L^{co-*} of L is $(\mathrm{IComp}(L))^*$. A language $L \subseteq \mathbb{M}(\mathcal{A})$ is *co-rational* if it can be constructed from finite languages in $\mathbb{M}(\mathcal{A})$ by union, product and concurrent iteration.

An automaton with concurrency relations $\mathcal{A}$ *forwardly preserves concurrency* if whenever $q.awba$ is defined and $a \parallel_q b$ then $a \parallel_{q.aw} b$ for any $a, b \in \Sigma$, $w \in \Sigma^*$ and $q \in Q$, see the following diagram.



Now the relationship between co-rational and recognizable languages in $\mathbb{M}(\mathcal{A})$ can be described as follows:

Theorem 5.1 ([21, 27, 48, 47]) *Let $\mathcal{A}$ be a finite automaton with concurrency relations.*

1. *If $\mathcal{A}$ forwardly preserves concurrency, then any recognizable language in $\mathbb{M}(\mathcal{A})$ is co-rational.*

2. *If $\mathcal{A}$ is stably concurrent, then a language in* $\mathrm{M}(\mathcal{A})$ *is co-rational if and only if it is recognizable.*

3. *Let $\mathcal{A}$ be stably concurrent. Then any rational language is recognizable if and only if there are no nonempty words u and v over Σ and state q such that $q.u = q.v = q$ and $u \parallel_q v$.*

The first statement of the above theorem and the implication $\Leftarrow$ of the second was shown in [21]. There, a weaker version of the remaining implication was proved as well. In an attempt to "algebraize" these proofs, we introduced divisibility monoids [27] where we could show a similar relation between recognizable and co-rational languages. In particular, the implication $\Rightarrow$ in the second statement follows in its full strength from this work. Even more, we show that it suffices to restrict to those co-rational constructions where the concurrent iteration is applied to languages K satisfying $K = \mathrm{IComp}(K)$. Similarly, the last statement follows from work on divisibility monoids, see [48, 47].

Let $\mathcal{A}$ be the automaton induced by the finite trace alphabet (Σ, I). Let furthermore $f : \mathrm{M}(\mathcal{A}) \setminus \{0\} \to M(\Sigma, I)$ be the canonical isomorphism generated by acseq. It is easy to see that a computation sequence $u \in \mathrm{CS}(\mathcal{A})$ is irreducible iff $f([u])$ is a connected trace. Hence, for $L \subseteq \mathrm{M}(\mathcal{A})$, $x \in \mathrm{IComp}(L)$ iff $f(x)$ is a connected component of an element of $f(L) \subseteq M(\Sigma, I)$. Since furthermore $\mathcal{A}$ is stably concurrent, Thm. 5.1 generalizes the well known result of Ochmański [59] on recognizable trace languages.

A fundamental result due to Büchi states that the recognizable languages in a free monoid over a finite alphabet are precisely those definable in monadic second order logic. This result has been extended to trace monoids by Thomas [69]. Now we show that this is also true for the monoids $\mathrm{M}(\mathcal{A})$ whenever $\mathcal{A}$ is stably concurrent.

Theorem 5.2 ([24]) *Let $\mathcal{A}$ be a finite stably concurrent automaton. Then $L \subseteq \mathrm{M}(\mathcal{A})$ is recognizable iff there exists a sentence $\varphi \in MSO$ such that $L = L(\varphi)$.*

The proof rests on a detailed analysis of the order-structure of $\mathrm{DO}(u)$ for $u \in \mathrm{CS}(\mathcal{A})$ as well as on Büchi's result. The use of particular accepting devices sometimes employed in the trace theoretic setting (*e.g.* asynchronous automata) is avoided.

In [23], the model of an asynchronous-cellular automaton (ACA) has been extended in such a way that it can accept arbitrary pomsets without autoconcurrency. Since computations of concurrent automata can be presented in such a way, these ACA are now capable of accepting languages in $\mathrm{M}(\mathcal{A})$. The following result, in its full generality, follows from [23] since the dependence orders of a given stably concurrent automaton are k-pomsets (this follows from [49]).

Theorem 5.3 ([23]) *Let $\mathcal{A}$ be a finite stably concurrent automaton. Then $L \subseteq \mathrm{M}(\mathcal{A})$ is recognizable iff there exists a finite asynchronous-cellular automaton $\mathcal{C}$ such that $L = L(\mathcal{C})$.*

In formal language theory, Büchi's original result [11, 10] characterizes the recognizability of a language of infinite words by the definability within monadic second order logic. Several authors considered an extension of this result to the realm of trace theory, which provides a natural framework for studying non-terminating concurrent systems (*e.g.* operating systems, transaction systems). In particular, Gastin, Petit and Zielonka [34] gave a Kleene-type characterization of the recognizable languages of infinite traces, and Ebinger and Muscholl [32] showed that these languages are precisely the ones definable by monadic second order logic, thereby extending Büchi's result.

Now we can consider languages $L \subseteq \mathrm{M}^\infty(\mathcal{A})$, *i.e.* languages of possibly infinity computations. Here it will be important that an infinite computation sequence $u = t_1 t_2 \ldots$ can also be seen as an infinite product of finite computation sequences t_i with $\operatorname{cod} t_i = \operatorname{dom} t_{i+1}$. Since the monoid-congruence $\sim$ on $\mathrm{CS}(\mathcal{A})$ is even a congruence with respect to infinite products, we can define infinite products of elements of $\mathrm{M}(\mathcal{A})$ by:

$$[u_1] \cdot [u_2] \cdot [u_3] \cdots = \begin{cases} [u_1 u_2 u_3 \ldots] & \text{if } \operatorname{cod} u_i = \operatorname{dom} u_{i+1} \text{ for } i \in \mathbb{N} \\ 0 & \text{otherwise.} \end{cases}$$

Hence, $\mathrm{M}^\infty(\mathcal{A})$ is the natural infinitary extension of $\mathrm{M}(\mathcal{A})$ together with an operation $\cdot : \mathrm{M}(\mathcal{A}) \times \mathrm{M}^\infty(\mathcal{A}) \to \mathrm{M}^\infty(\mathcal{A})$.

As usual, if $U \subseteq \mathrm{M}(\mathcal{A})$ and $V \subseteq \mathrm{M}^\infty(\mathcal{A})$, we put $U \cdot V = \{u \cdot v \mid u \in U, v \in V\}$ and let U^ω be the set of all infinite products $x_1 \cdot x_2 \cdot x_3 \ldots$ with $x_i \in U$ for $i \in \mathbb{N}$.

A language $L \subseteq \mathrm{M}^\infty(\mathcal{A})$ is *recognizable* if there exists a finite monoid $(S, \cdot)$ and a homomorphism $\eta : M \to S$ such that for any sequence $(x_i)_{i<\omega}$ of finite (possibly empty) computations with $x_1 \cdot x_2 \cdot x_3 \cdots \in L$ we have $\eta^{-1}\eta(x_1) \cdot \eta^{-1}\eta(x_2) \cdot \eta^{-1}\eta(x_3) \cdots \subseteq L$. It can be checked that a language $L \subseteq \mathrm{M}(\mathcal{A})$ is recognizable in $\mathrm{M}(\mathcal{A})$ iff it is recognizable in $\mathrm{M}^\infty(\mathcal{A})$. The recognizable languages in $\mathrm{M}^\infty(\mathcal{A})$ are completely described by the following result:

Theorem 5.4 ([26]) *Let $\mathcal{A}$ be a finite stably concurrent automaton and $L \subseteq \mathrm{M}^\infty(\mathcal{A})$. Then the following are equivalent.*

1. *L is recognizable.*
2. *L is a Boolean combination of languages $U \cdot V^\omega$ where $U, V \subseteq \mathrm{M}(\mathcal{A})$ are recognizable and $V \cdot V \subseteq V$.*
3. *There exists a sentence $\varphi \in$MSO such that $L = L^\infty(\varphi)$.*

The equivalence of 1 and 3 in this theorem generalizes the result of Ebinger and Muscholl mentioned above. It also contains Thm. 5.2 (which, however, is used for its proof). The result of Gastin, Petit and Zielonka does not follow in its whole strength from the equivalence of 1 and 2, but at least we obtained a Kleene-type characterization of recognizable languages in $\mathrm{M}^\infty(\mathcal{A})$, since the recognizable languages of $\mathrm{M}(\mathcal{A})$ are characterized in this way by Thm. 5.1. In this sense Thm. 5.4 extends the result of [34].

6 Aperiodic languages in $\mathrm{M}(\mathcal{A})$

A language L in a monoid M is *aperiodic* if there exists a finite aperiodic monoid $(S, \cdot)$ and a homomorphism $\eta : M \to S$ such that $\eta^{-1}\eta(L) = L$; here $(S, \cdot)$ is *aperiodic* if there exists $n \in \mathbb{N}$ such that $s^{n+1} = s^n$ for any $s \in S$. Clearly, any aperiodic language is recognizable. A language $L \subseteq M$ is *starfree* if it can be constructed from finite languages in M by Boolean operations and product.

For a free monoid $M = \Sigma^\star$ (Σ finite), Schützenberger [64] proved that a language $L \subseteq \Sigma^\star$ is aperiodic iff it is starfree. This result has been extended by Guaiana, Restivo and Salemi [38] to the case of trace monoids where again these language classes are equal. This does not remain true for the monoid $\mathrm{M}(\mathcal{A})$ (even if $\mathcal{A}$ is stably concurrent). But in $\mathrm{M}(\mathcal{A})$, any aperiodic language is starfree. We say that $\mathcal{A}$ *has no commuting counters*, if $a \parallel_q w^n$ and $q.w^n = q$ imply $q.w = q$ for any $a \in \Sigma$, $w \in \Sigma^\star$ and $q \in Q$. Now we have

Theorem 6.1 ([22]) *Let $\mathcal{A}$ be a finite stably concurrent automaton without commuting counters. Then $L \subseteq \mathrm{M}(\mathcal{A})$ is aperiodic iff it is starfree.*

Observe that an automaton induced by a trace alphabet has no commuting counters (since $q.w = q$ for any $w \in \Sigma^\star$). Hence this result generalizes the result of Guaiana, Restivo and Salemi.

We now turn to the definability of aperiodic languages by logical means similar to Thm. 5.2. McNaughton and Papert [55] showed that the aperiodic and the first order definable languages in a finitely generated free monoid coincide. This result has been extended to trace monoids by Thomas [69] and by Ebinger and Muscholl [32]. For finite stably concurrent automata $\mathcal{A}$, the classes of aperiodic and of first-order definable languages in $\mathrm{M}(\mathcal{A})$ are incomparable (see [24] for an example). Therefore, we again need additional assumptions on $\mathcal{A}$.

A stably concurrent automaton $\mathcal{A}$ is *counter-free* if $q.w^n = q$ implies $q.w = q$ for any $q \in Q$ and $w \in \Sigma^\star$. $\mathcal{A}$ has *no commuting loops* if $a \parallel_q w$ implies $q.w \neq q$ for any $a \in \Sigma$, $w \in \Sigma^\star$ and $q \in Q$. It is an *automaton with global independence* if whenever $a \parallel_p b$ and $q.ab$ is defined then $a \parallel_q b$ for any $a, b \in \Sigma$ and $p, q \in Q$.

Again, observe that the automata induced by trace alphabets satisfy all these conditions. Hence the following results generalize corresponding results on trace monoids.

Theorem 6.2 ([24])

1. *Let $\mathcal{A}$ be a finite counter-free stably concurrent automaton. Let $L \subseteq \mathrm{M}(\mathcal{A})$ be aperiodic. Then there exists a sentence $\varphi \in FO$ such that $L = L(\varphi)$.*
2. *Let $\mathcal{A}$ be a finite stably concurrent automaton without commuting loops or with global independence. Let φ be a sentence of FO. Then $L(\varphi)$ is aperiodic.*

Several temporal logics have been considered for traces. The first expressively complete one was Ebinger's TLPO [31] (see [25] for its generalization to stably concurrent automata). This logic is expressively complete for finite computations and uses both, past and future modalities. Thiagarajan and Walukiewicz defined the temporal logic LTL for traces [68]; it is expressively complete for finite and infinite computations and it is a pure future logic. Unfortunately, its satisfiability problem is nonelementary [70]. The logic LTL can also be extended to stably concurrent automata along the lines of [68]. Since concurrent automata generalize traces, this nonelementary lower bound holds in this case as well – but we can show that under suitable restrictions on the automaton, the problem becomes elementary:

Theorem 6.3 *Let $\mathcal{A}$ be a finite stably concurrent automaton and $n \in \mathbb{N}$ such that for any $q \in Q$ and $u, v \in \Sigma^+$ we have*

$$u \parallel_q v, |u| > n \Rightarrow |v| < n.$$

Then the satisfiability problem "$L(\varphi) = \emptyset$?" is solvable in EXPTIME for formulas of the temporal logic LTL.

References

[1] P. Bachmann and Phan Minh Dung. Nondeterministic computations - structure and axioms. *Elekton. Inform.-verarb. Kybern. EIK*, 22:243–261, 1986.

[2] G. Bednarczyk. *Categories of asynchronous systems.* PhD thesis, University of Sussex, 1987.

[3] G. Berry and J.-J. Levy. Minimal and optimal computations of recursive programs. *J. ACM*, 26:148–175, 1979.

[4] G. Birkhoff. *Lattice Theory*. Colloquium Publications vol. 25. American Mathematical Society, Providence, 1940. Page numbers refer to the third edition, seventh printing from 1993.

[5] P. Boldi, F. Cardone, and N. Sabadini. Concurrent automata, prime event structures and universal domains. In M. Droste and Y. Gurevich, editors, *Semantics of Programming Languages and Model Theory*, pages 89–108. Gordon and Breach Science Publ., OPA Amsterdam, 1993.

[6] G. Boudol. Computational semantics of term rewriting. In M. Nivat and J.C. Reynolds, editors, *Algebraic Methods in Semantics*, pages 169–236. Cambridge University Press, 1985.

[7] G. Boudol and I. Castellani. A non-interleaving semantics for CCS based on proved transitions. *Fundam. Inform.*, 11:433–452, 1988.

[8] F. Bracho and M. Droste. Labelled domains and automata with concurrency. *Theoretical Comp. Science*, 135:289–318, 1994.

[9] F. Bracho, M. Droste, and D. Kuske. Representation of computations in concurrent automata by dependence orders. *Theoretical Comp. Science*, 174:67–96, 1997.

[10] J.R. Büchi. On a decision method in restricted second order arithmetics. In E. Nagel et al., editors, *Proc. Intern. Congress on Logic, Methodology and Philosophy of Science*, pages 1–11. Stanford University Press, Stanford, 1960.

[11] J.R. Büchi. Weak second-order arithmetic and finite automata. *Z. Math. Logik Grundlagen Math.*, 6:66–92, 1960.

[12] P. Cartier and D. Foata. *Problemes combinatoires de commutation et rearrangements*. Lecture Notes in Mathematics vol. 85. Springer, Berlin - Heidelberg - New York, 1969.

[13] P.L. Curien. *Categorical Combinators, Sequential Algorithms and Functional Programming*. Progress in Theor. Comp. Science. Birkhäuser Boston, 1993.

[14] P. Dehornoy. On completeness of word reversing. *Discrete Mathematics* (special issue on Formal power series and algebraic combinatorics, Toronto, 1998), 225:93–119, 2000.

[15] P. Dehornoy. Complete positive group presentations. Technical Report 2001-13, University of Caen, 2001.

[16] V. Diekert. *Combinatorics on Traces*. Lecture Notes in Comp. Science vol. 454. Springer, 1990.

[17] V. Diekert. On the concatenation of infinite traces. *Theoretical Comp. Science*, 113:35–54, 1993.

[18] V. Diekert and G. Rozenberg. *The Book of Traces*. World Scientific Publ. Co., 1995.

[19] M. Droste. Concurrency, automata and domains. In *17th ICALP*, Lecture Notes in Comp. Science vol. 443, pages 195–208. Springer, 1990.

[20] M. Droste. Concurrent automata and domains. *Intern. J. of Found. of Comp. Science*, 3:389–418, 1992.

[21] M. Droste. Recognizable languages in concurrency monoids. *Theoretical Comp. Science*, 150:77–109, 1995.

[22] M. Droste. Aperiodic languages in concurrency monoids. *Information and Computation*, 126:105–113, 1996.

[23] M. Droste, P. Gastin, and D. Kuske. Asynchronous cellular automata for pomsets. *Theoretical Comp. Science*, 247:1–38, 2000. (Fundamental study).

[24] M. Droste and D. Kuske. Logical definability of recognizable and aperiodic languages in concurrency monoids. In *Computer Science Logic*, Lecture Notes in Comp. Science vol. 1092, pages 467–478. Springer, 1996.

[25] M. Droste and D. Kuske. Temporal logic for computations of concurrent automata. Technical Report MATH-AL-3-1996, Inst. für Algebra, TU Dresden, 1996.

[26] M. Droste and D. Kuske. Recognizable and logically definable languages of infinite computations in concurrent automata. *International Journal of Foundations of Computer Science*, 9:295–313, 1998.

[27] M. Droste and D. Kuske. Recognizable languages in divisibility monoids. *Mathematical Structures in Computer Science*, 11:741–770, 2001.

[28] M. Droste and R.M. Shortt. Petri nets and automata with concurrency relations - an adjunction. In M. Droste and Y. Gurevich, editors, *Semantics of Programming Languages and Model Theory*, pages 69–87. Gordon and Breach Science Publ., OPA, Amsterdam, 1993.

[29] M. Droste and R.M. Shortt. Continuous Petri nets and transition systems. In H. Ehrig, G. Juhas, J. Padberg and G. Rozenberg, editors, *Unifying Petri Nets*, Lecture Notes in Comp. Science vol. 2128, pages 457–484, 2001.

[30] M. Droste and R.M. Shortt. From Petri nets to automata with concurrency. *Applied Categorical Structures*, 10:173–191, 2002.

[31] W. Ebinger. *Charakterisierung von Sprachklassen unendlicher Spuren durch Logiken*. PhD thesis, Universität Stuttgart, 1994.

[32] W. Ebinger and A. Muscholl. Logical definability on infinite traces. *Theoretical Comp. Science*, 154:67–84, 1996.

[33] S. Eilenberg. *Automata, Languages and Machines vol. A*. Academic Press, New York, 1974.

[34] P. Gastin, A. Petit, and W. Zielonka. An extension of Kleene's and Ochmanski's theorems to infinite traces. *Theoretical Comp. Science*, 125:167–204, 1994.

[35] P. Godefroid. *Partial-Order Methods for the Verification of Concurrent Systems*. Lecture Notes in Comp. Science vol. 1032. Springer, 1996.

[36] P. Godefroid, D. Peled, and M. Staskauskas. Using partial order methods in the formal validation of industrial concurrent programs. *IEEE Transactions on Software Engineerings*, 22:496–507, 1996.

[37] P. Godefroid and D. Pirottin. Refining dependencies improves partial-order verification methods. In *CAV'93*, Lecture Notes in Comp. Science vol. 697, pages 438–449, 1993.

[38] G. Guaiana, A. Restivo, and S. Salemi. Star-free languages. *Theoretical Comp. Science*, 97:301–311, 1992.

[39] C.A.R Hoare. Communicating sequential processes. *Commun. ACM*, 21:666–676, 1978.

[40] G. Huet and J.-J Levy. Call-by-need computations in non-ambiguous linear term rewriting systems. Technical report, IRIA-LABORIA Report 359, Paris, 1979.

[41] A. Jung. *Cartesian Closed Categories of Domains*. CWI Tract no. 66. Amsterdam, 1989.

[42] S. Katz and D. Peled. Defining conditional independence using collapses. In M.Z. Kwiatkowska, M.W. Shields, and R.M. Thomas, editors, *Semantics for Concurrency, Proc. of the Int. BCS-FACS Workshop at Leicester*, pages 262–280. Springer, 1990.

[43] S. Katz and D. Peled. Defining conditional independence using collapses. *Theoretical Comp. Science*, 101:337–359, 1992.

[44] D. Kuske. *Modelle nebenläufiger Prozesse – Monoide, Residuensysteme und Automaten*. PhD thesis, Universität GHS Essen, 1994.

[45] D. Kuske. Representation of domains by residuum systems. In *Workshop Domains*, pages 91–98. TH Darmstadt, 1994.

[46] D. Kuske. Symmetries of the partial order of traces. *Order*, 16:133–148, 1999.

[47] D. Kuske. Contributions to a Trace Theory beyond Mazurkiewicz Traces. Habilitationsschrift, TU Dresden, 2000.

[48] D. Kuske. Divisibility monoids: presentation, word problem, and rational languages. In R. Freivalds, editor, *FCT 2001*, Lecture Notes in Comp. Science vol. 2138, pages 227–239. Springer, 2001.

[49] D. Kuske and R. Morin. Pomsets for local trace languages: Recognizability, logic and Petri nets. *Journal of Automata, Languages and Combinatorics*, 2002. To appear.

[50] D. Kuske and R.M. Shortt. Topology for computations of concurrent automata. *International Journal of Algebra and Computation*, 8:327–362, 1998.

[51] M. Kwiatkowska. A metric for traces. *Information Processing Letters*, 35:129–135, 1990.

[52] J.-J. Levy. Optimal reductions in the lambda calculus. In J.P. Seldin and J.R. Hindley, editors, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 159–191. Academic Press, New York, 1980.

[53] A. Mazurkiewicz. Concurrent program schemes and their interpretation. Technical report, DAIMI Report PB-78, Aarhus University, 1977.

[54] A. Mazurkiewicz. Traces theory. In W. Brauer ant others, editor, *Petri Nets, Applications and Relationship to other Models of Concurrency*, Lecture Notes in Comp. Science vol. 255, pages 279–324. Springer, 1987.

[55] R. McNaughton and S. Papert. *Counter-Free Automata*. MIT Press, Cambridge, USA, 1971.

[56] R. Milner. *Calculus of Communicating Processes*. Lecture Notes in Comp. Science vol. 92. Springer, 1980.

[57] M. Mukund. Petri nets and step transition systems. *International Journal of Foundations of Computer Science*, 3:443–478, 1992.

[58] M. Nielsen, G. Rosenberg, and P.S. Thiagarajan. Elementary transition systems. *Theoretical Comp. Science*, 96:3–33, 1992.

[59] E. Ochmański. Regular behaviour of concurrent systems. *Bull. Europ. Assoc. for Theor. Comp. Science*, 27:56–67, 1985.

[60] P. Panangaden, V. Shanbhogue, and E.W. Stark. Stability and sequentiality in dataflow networks. In *17th ICALP*, Lecture Notes in Comp. Science vol. 443, pages 308–321. Springer, 1990.

[61] P. Panangaden and E.W. Stark. Computations, residuals and the power of indeterminacy. In *Automata, Languages and Programming*, Lecture Notes in Comp. Science vol. 317, pages 439–454. Springer, 1988.

[62] M. Picantin. The center of thin gaussian groups. *J. of Algebra*, 245:92–122, 2001.

[63] W. Reisig. *Petri Nets*. Springer, 1985.

[64] M.P. Schützenberger. On finite monoids having only trivial subgroups. *Inf. Control*, 8:190–194, 1965.

[65] M.W. Shields. Concurrent machines. *Comp. J.*, 28:449–465, 1985.

[66] E.W. Stark. Concurrent transition systems. *Theoretical Comp. Science*, 64:221–269, 1989.

[67] E.W. Stark. Connections between a concrete and an abstract model of concurrent systems. In *5th Conf. on the Mathematical Foundations of Programming Semantics*, Lecture Notes in Comp. Science vol. 389, pages 53–79. Springer, 1989.

[68] P.S. Thiagarajan and I. Walukiewicz. An expressively complete linear time temporal logic for Mazurkiewicz traces. In *LICS'97*, pages 183–194. IEEE Computer Society Press, 1997. (full version to appear in *Information and Computation*).

[69] W. Thomas. On logical definability of trace languages. In V. Diekert, editor, *Proceedings of a workshop of the ESPRIT BRA No 3166: Algebraic and Syntactic Methods in Computer Science (ASMICS) 1989*, Report TUM-I9002, Technical University of Munich, pages 172–182, 1990.

[70] I. Walukiewicz. Difficult configurations – on the complexity of LTrL. In *ICALP'98*, Lecture Notes in Comp. Science vol. 1443, pages 140–151. Springer, 1998.

[71] G. Winskel. Event structures. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri nets: Applications and Relationships to Other Models of Concurrency*, Lecture Notes in Comp. Science vol. 255, pages 325–392. Springer, 1987.

[72] G. Winskel and M. Nielsen. Models for concurrency. In S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science vol. 4*, pages 1–148. Oxford University Press, 1994.

[73] W. Zielonka. Notes on finite asynchronous automata. *R.A.I.R.O. - Informatique Théorique et Applications*, 21:99–135, 1987.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Learning with Skewed Class Distributions

Maria Carolina Monard and Gustavo E. A. P. A. Batista
University of São Paulo - USP
Institute of Mathematics and Computer Science - ICMC
Department of Computer Science and Statistics - SCE
Laboratory of Computational Intelligence - LABIC
P. O. Box 668, 13560-970 - São Carlos, SP, Brazil
{gbatista, mcmonard}@icmc.sc.usp.br

Abstract. Several aspects may influence the performance achieved by a classifier created by a Machine Learning system. One of these aspects is related to the difference between the numbers of examples belonging to each class. When this difference is large, the learning system may have difficulties to learn the concept related to the minority class. In this work, we discuss several issues related to learning with skewed class distributions, such as the relationship between cost-sensitive learning and class distributions, and the limitations of accuracy and error rate to measure the performance of classifiers. Also, we survey some methods proposed by the Machine Learning community to solve the problem of learning with imbalanced data sets, and discuss some limitations of these methods.

1 Introduction

Supervised learning is the process of automatically creating a classification model from a set of examples, called the *training set*, which belong to a set of classes. Once a model is created, it can be used to automatically predict the class of other unclassified examples.

In other words, in supervised learning, a set of n training examples is given to an inducer. Each example $\mathbf{X}$ is an element of the set $F_1 \times F_2 \times \ldots \times F_m$ where F_j is the domain of the jth feature. Training examples are tuples $(\mathbf{X}, Y)$ where Y is the label, output or class. The Y values are typically drawn from a discrete set of classes $\{1, \ldots, K\}$ in the case of *classification*. Given a set of training examples, the learning algorithm (*inducer*) outputs a *classifier* such that, given a new example, it accurately predicts the label Y.

In this work, we reserve our discussion to concept-learning[1], so Y can assume one of two mutually exclusive values. We use the general labels **positive** and **negative** to discriminate between the two class values.

For a number of application domains, a huge disproportion in the number of cases belonging to each class is common. For instance, in detection of fraud in telephone calls [7] and credit card transactions [15], the number of legitimate transactions is much higher than the number of fraudulent transactions. In insurance risk modelling [12], only a small percentage of the policyholders file one or more claims in any given time period. Also, in direct marketing [11], it is common to have a small response rate (about 1%) for most marketing campaigns. Other examples of domains with intrinsic imbalance can be found in the literature. Thus, learning with skewed class distributions is an important issue in supervised learning.

---

[1] However some of the methods discussed here can be applied to multi-class problems.

Many traditional learning systems are not prepared to induce a classifier that accurately classifies the minority class under such situation. Frequently, the classifier has a good classification accuracy for the majority class, but its accuracy for the minority class is unacceptable. The problem arises when the misclassification cost for the minority class is much higher than the misclassification cost for the majority class. Unfortunately, that is the norm for most applications with imbalanced data sets, since these applications aim to profile a small set of valuable entities that are spread in a large group of "uninteresting" entities.

In this work we discuss some of the most used methods that aim to solve the problem of learning with imbalanced data sets. These methods can be divided into three groups:

1. **Assign misclassification costs.** In a general way, misclassify examples of the minority class is more costly than misclassify examples of the majority class. The use of cost-sensitive learning systems might aid to solve the problem of learning from imbalanced data sets;

2. **Under-sampling.** One very direct way to solve the problem of learning from imbalanced data sets is to artificially balance the class distributions. Under-sampling aim to balance a data set by eliminating examples of the majority class;

3. **Over-sampling.** This method is similar to under-sampling. But it aims to achieve a more balanced class distributions by replicating examples of the minority class.

This work is organised as follows: Section 2 discusses why accuracy and error rate are inadequate metrics to measure the performance of learning systems when data have asymmetric misclassification costs and/or class imbalance; Section 3 explains the relationship between imbalanced class distributions and cost-sensitive learning; Section 4 discusses which class distributions are best for learning; Section 5 surveys some methods proposed by the Machine Learning community to balance the class distributions; Section 6 presents a brief discussion about some evidences that balancing a class distributions has little effect in the final classifier; finally, Section 7 shows the conclusions of this work.

2 Why Accuracy and Error Rate are Inadequate Performance Measures for Imbalanced Data Sets

Different types of errors and hits performed by a classifier can be summarised in a *confusion matrix*. Table 1 illustrates a confusion matrix for a two class problem, with classes labelled **positive** and **negative**:

From such matrix it is possible to extract a number of metrics to measure the performance of learning systems, such as error rate $E = \frac{(c+b)}{(a+b+c+d)}$ and accuracy $Acc = \frac{(a+d)}{(a+b+c+d)} = 1 - E$.

	Positive Prediction	*Negative Prediction*
Positive Class	True Positive (a)	False Negative (b)
Negative Class	False Positive (c)	True Negative (d)

Table 1: Different types of errors and hits for a two classes problem.

The error rate (E) and the accuracy (Acc) are widely used metrics for measuring the performance of learning systems. However, when the prior probabilities of the classes are very different, such metrics might be misleading. For instance, it is straightforward to create a classifier having 99% accuracy (or 1% error rate) if the data set has a majority class with 99% of the total number of cases, by simply labelling every new case as belonging to the majority class.

Other fact against the use of accuracy (or error rate) is that these metrics consider different classification errors as equally important. For instance, a sick patience diagnosed as healthy might be a fatal error while a healthy patience diagnosed as sick is considered a much less serious error since this mistake can be corrected in future exams. On domains where misclassification cost is relevant, a cost matrix could be used. A cost matrix defines the misclassification cost, i.e., a penalty for making a mistake for each different type of error. In this case, the goal of the classifier is to minimize classification cost instead of error rate. Section 3 discusses more about the relationship between cost-sensitive learning and imbalanced data sets.

It would be more interesting if we could use a performance metric that disassociates the errors (or hits) occurred in each class. From Table 1 it is possible to derive four performance metrics that directly measure the classification performance on the positive and negative classes independently, they are:

- **False negative rate**: $FN = \frac{b}{a+b}$ is the percentage of positive cases misclassified as belonging to the negative class;
- **False positive rate**: $FP = \frac{c}{c+d}$ is the percentage of negative cases misclassified as belonging to the positive class;
- **True negative rate**: $TN = \frac{d}{c+d} = 1 - FP$ is the percentage of negative cases correctly classified as belonging to the negative class;
- **True positive rate**: $TP = \frac{a}{a+b} = 1 - FN$ is the percentage of positive cases correctly classified as belonging to the positive class;

These four class performance measures have the advantage of being independent of class costs and prior probabilities. It is obvious that the main objective of a classifier is to minimize the false positive and negative rates or, similarly, to maximize the true negative and positive rates. Unfortunately, for most "real world" applications, there is a tradeoff between FN and FP and, similarly, between TN and TP. The *ROC*[2] *graphs* [13] can be used to analyse the relationship between FN and FP (or TN and TP) for a classifier.

---

[2] ROC is an acronym for *Receiver Operating Characteristic*, a term used in signal detection to characterize the tradeoff between hit rate and false alarm rate over a noisy channel.

Consider that the minority class, whose performance will be analysed, is the positive class. On a ROC graph, TP $(1 - FN)$ is plotted on the Y axis and FP is plotted on the X axis. Some classifiers have parameter for which different settings produce different ROC points. For instance, a classifier that produces probabilities of an example being in each class, such as Naive Bayes classifier, can have a threshold parameter biasing the final class selection[3]. Plotting all the ROC points that can be produced by varying these parameters produces a ROC curve for the classifier. Typically this is a discrete set of points, including (0,0) and (1,1), which are connected by line segments. Figure 1 illustrates a ROC graph of 3 classifiers: A, B and C. Several points on a ROC graph should be noted. The lower left point (0,0) represents a strategy that classifies every example as belonging to the negative class. The upper right point represents a strategy that classifies every example as belonging to the positive class. The point (0,1) represents the perfect classification, and the line $x = y$ represents the strategy of random guessing the class.

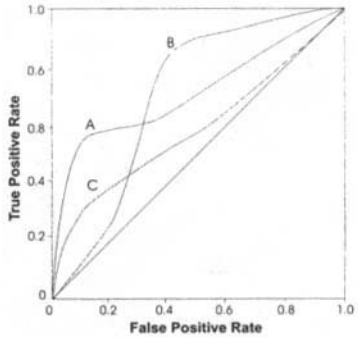


Figure 1: A ROC graph for 3 classifiers.

From a ROC graph is possible to calculate an overall measure of quality, the *under the ROC curve area (AUC)*. The AUC is the fraction of the total area that falls under the ROC curve. This measure is equivalent to several other statistical measures for evaluating classification and ranking models [8]. The AUC effectively factors in the performance of the classifier over all costs and distributions.

3 Imbalance and Cost-Sensitive Learning

A classifier induced from an imbalanced data set has, typically, a low error rate for the majority class and an unacceptable error rate for the minority class. The problem arises when the misclassification cost for the minority class is much higher than the misclassification cost for the majority class. In this situation, it is important to accurately classify the minority class in order to reduce the overall cost.

---

[3] In a similar way, other learning system can be adapted to produce such posterior probabilities estimates. In decision trees, the class distributions at the leaves can be use as an estimate. Rule learning systems can make similar estimates. Neural networks produce continuous outputs that can be mapped to probability estimates.

A cost-sensitive learning system can be used in applications where the misclassification costs are known. Cost-sensitive learning systems attempt to reduce the cost of misclassified examples, instead of classification error.

Unfortunately, some learning systems are not able to integrate cost information into the learning process. However, there is a simple and general method to make any learning system cost-sensitive for a concept-learning problem [2]. The idea is to change the class distributions in the training set towards the most costly class. Suppose that the positive class is five times more costly than the negative class. If the number of positive examples are artificially increased by a factor of five, then the learning system, aiming to reduce the number of classification errors, will come up with a classifier that is skewed towards the avoidance of error in the positive class, since any such errors are penalised 5 times more. In [6] is provided a theorem that shows how to change the proportion of positive and negative examples in order to make optimal cost-sensitive classifications for a concept-learning problem. Moreover, a general method to make learning system cost-sensitive is presented in [4]. This method has the advantage of being applicable to multi-class problems.

Thus, class imbalance and cost-sensitive learning are related to each other. One way to learn with an imbalanced data set is to train a cost-sensitive learning system with the misclassification cost of the minority class greater than the majority class. One way to make a learning system cost-sensitive is to intentionally imbalance the training set.

Most methods that deal with imbalanced data sets aim to improve the learning of the minority concept by balancing the data set. In this way, the minority class becomes more costly, and we can expect it will be better classified. From this point, two different situations can be identified: first, there are plenty of data, and the problem can be understood as "which proportion of positive/negative examples is the best for learning?"; second, data are scarce, and there is another additional problem: "how discard negative examples/duplicate positive examples without introducing much bias in the learning process?". In the following sections these two questions are discussed in more detail.

4 Which Proportion of Positive/Negative Examples is the Best for Learning?

The problem of determining which proportion of positive/negative examples is the best for learning goes beyond the problem of learning from imbalanced data sets. Much of the Machine Learning community has assumed that the naturally occurring class distributions are the best for learning. However, because of leaning from the naturally occurring class distributions yield bad results for highly imbalanced data sets, this assumption started to be studied in deep.

In [17] is made a throughout study about the proportion of positive/negative examples in learning. This study analyses the scenario that there are plenty of data, however because of computational restriction, the training set should be limited to n examples. In this scenario, which class distributions should be the best for training?

Using the area under the ROC curve (AUC) as performance measure, [17] shows that the optimal distribution generally contains between 50% and 90% of minority class examples. Also, allocating 50% of the training examples to the minority class, while it will not always yield optimal results, will generally lead to results which are no worse than, and often superior to, those which use the natural class distributions.

5 How Discard Negative Examples/Duplicate Positive Examples Without Introducing Much Bias in the Learning Process?

One of the most direct ways for dealing with class imbalance is to alter the class distributions toward a more balanced distribution. There are two basic methods for balancing the class distributions:

1. **Under-sampling**: these methods aim to balance the data set by eliminating examples of the majority class, and;
2. **Over-sampling**: these methods replicate examples of the minority class in order to achieve a more balanced distribution.

Both, under-sampling and over-sampling, have known drawbacks. Under-sampling can throw away potentially useful data, and over-sampling can increase the likelihood of occurring overfitting, since most of over-sampling methods make exact copies of the minority class examples. In this way, a symbolic classifier, for instance, might construct rules that are apparently accurate, but actually, cover one replicated example.

Some recent research has focused in overcoming the drawbacks of both under-sampling and over-sampling. In [3] under-sampling and over-sampling methods are combined, and, instead of over-sampling by replicating minority class examples, new minority class examples are formed by interpolating between several minority class examples that lie together. Thus, they avoid the overfitting problem and cause the decision boundaries for the minority class to spread further into the majority class space.

In [10, 1] an under-sampling technique is analysed in order to minimize the amount of potentially useful data discarded. The majority class examples are classified as "safe", "borderline" and "noise" examples. Borderline and noisy cases are detected using *Tomek links* [16], and are removed from the data set. Only safe majority class examples and all minority class examples are used for training the learning system. A Tomek link can be defined as follows: given two examples x and y belonging to different classes, and be $d(x, y)$ the distance between x and y. A (x, y) pair is called a Tomek link if there is not a case z, such that $d(x, z) < d(x, y)$ or $d(y, z) < d(y, x)$. Figure 2 illustrates the process of cleaning a data set with Tomek links.

Other methods for reducing the training set size based on k-nearest neighbor, like Tomek links, are surveyed in [18].

6 Discussion

Much of research done to solve the problem of learning from imbalanced data sets is based on balancing the class distributions. However, recent research has shown that many learning systems are insensitive to class distributions. Drummond and Holte [5] showed that there are decision tree splitting criteria that are relatively insensitive to a data set class distributions. Elkan [6] makes similar statements for Naive Bayes and

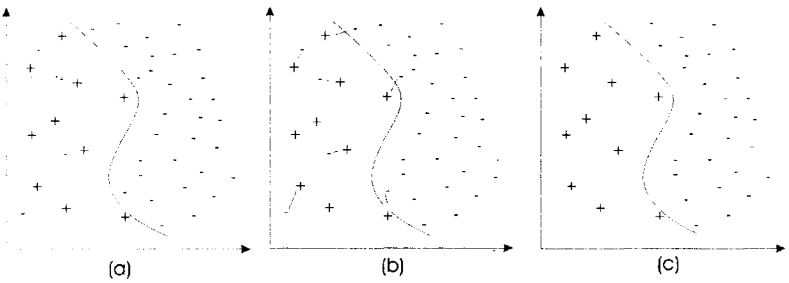


Figure 2: Applying Tomek links to a data set. Original data set *(a)*, Tomek links identified *(b)*, and Tomek links removed *(c)*.

decision tree learning systems. If a learning system is insensitive to the class distributions, then changing the class distributions — or balancing a data set — might have little effect in the induced classifier.

On the other hand, under- and over-sampling have been empirically analysed in several of domains, with good results. In [9] several approaches for dealing with imbalanced data sets are compared, and it concludes that under- and over-sampling are very effective methods for dealing with imbalanced data sets.

Moreover, Drummond and Holte [5] stated that under- and over-sampling should be reconsidered in terms of how they affect pruning and leaf labelling. However, on several experiments performed in [14], classifiers generated from balanced distributions obtained results that were, frequently, better than those obtained from the naturally occurring distributions. These experiments were conducted with no pruning, and adjusting the leaf labelling to account the changes made in class distributions.

7 Conclusion

Learning from imbalanced data sets is an important issue in Machine Learning. A direct method to solve the imbalance problem is to artificially balance the class distributions. This balance can be obtained by under-sampling the majority class, over-sampling minority class, or both. There are several works in the literature that confirm the efficiency of these methods in practice. However, there is some evidence that re-balancing the class distributions artificially does not have much effect on the performance of the induced classifier, since some learning systems are not sensitive to differences in class distributions. It seems that we still need a clearer understanding of how class distributions affect each phase of the learning process. For instance, in decision trees, how class distributions affect the tree construction, pruning and leaf labelling. A deeper understanding of the basics will permit us to design better methods for dealing with the problem of learning with skewed class distributions.

Acknowledgements. This research is partially supported by Brazilian Research Councils CAPES and FINEP.

References

[1] G. E. A. P. A. Batista, A. Carvalho, and M. C. Monard. Applying One-sided Selection to Unbalanced Datasets. In O. Cairo, L. E. Sucar, and F. J. Cantu, editors, *Proceedings of the Mexican International Conference on Artificial Intelligence - MICAI 2000*, pages 315–325. Springer-Verlag, April 2000. Best Paper Award Winner.

[2] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth & Books, Pacific Grove, CA, 1984.

[3] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Jounal of Artificial Intelligence Research*, 16:321–357, 2002.

[4] Pedro Domingos. MetaCost: A General Method for Making Classifiers Cost-Sensitive. In *Knowledge Discovery and Data Mining*, pages 155–164, 1999.

[5] Chris Drummond and Robert C. Holte. Exploiting the Cost (In)sensitivity of Decision Tree Splitting Criteria. In *Proceedings of the 17th International Conference on Machine Learning (ICML'2000)*, pages 239-246, 2000.

[6] Charles Elkan. The Foundations of Cost-Sensitive Learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 973–978, 2001.

[7] Tom Fawcett and Foster J. Provost. Adaptive Fraud Detection. *Data Mining and Knowledge Discovery*, 1(3):291–316, 1997.

[8] David J. Hand. *Construction and Assessment of Classification Rules*. John Wiley and Sons, 1997.

[9] Nathalie Japkowicz. Learning from Imbalanced Data Sets: a Comparison of Various Strategies. In *AAAI Workshop on Learning from Imbalanced Data Sets*, Menlo Park, CA, 2000. AAAI Press.

[10] M. Kubat and S. Matwin. Addressing the Course of Imbalanced Training Sets: One-Sided Selection. In *Proceedings of 14th International Conference in Machine Learning*, pages 179–186, San Francisco, CA, 1997. Morgan Kaufmann.

[11] Charles X. Ling and Chenghui Li. Data Mining for Direct Mining: Problems and Solutions. In *Proceedings of The Forth International Conference on Knownledge Discovery and Data Mining*, pages 73–79, 1998.

[12] Edwin P. D. Pednault, Barry K. Rosen, and Chidanand Apte. Handling Imbalanced Data Sets in Insurance Risk Modeling. Technical Report RC-21731, IBM Research Report, March 2000.

[13] Foster J. Provost and Tom Fawcett. Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions. In *Knowledge Discovery and Data Mining*, pages 43–48, 1997.

[14] Foster J. Provost and Tom Fawcett. Robust Classification for Imprecise Environments. *Machine Learning*, 42(3):203–231, 2001.

[15] S. J. Stolfo, D. W. Fan, W. Lee, A. L. Prodromidis, and P. K. Chan. Credit Card Fraud Detection Using Meta-Learning: Issues and Initial Results. In *AAAI-97 Workshop on AI Methods in Fraud and Risk Management*, 1997.

[16] I. Tomek. Two Modifications of CNN. *IEEE Transactions on Systems Man and Communications*, SMC-6:769–772, 1976.

[17] Gary M. Weiss and Foster Provost. The Effect of Class Distribution on Classifier Learning: An Empirical Study. Technical Report ML-TR-44, Rutgers University, Department of Computer Science, 2001.

[18] D. R. Wilson and T. R. Martinez. Reduction Techniques for Exemplar-Based Learning Algorithms. *Machine Learning*, 38(3):257–286, March 2000.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

An Enlargement of Theorems for Sentential Calculus

Shotaro Tanaka
Tokushima Bunri University
Shido,Sanuki,Kagawa,769-2193,Japan

Abstract. We shall enlarge the Sentential Calculus by Lukasiewicz. We use his symbolizs,and the concept of 'yield' by Rosser. I introduced a deduction theorem,which we call a con-junctive deduction theorem. Here we shall prove new theorems and theses. Theorems areexpressed with 'yield' i.e., , Γ:a set of propositions',and theses are va-lid propositions in the system. $\vdash$

1. Two axiom systems.

$\boxed{1}$ — 1 Lukasiewicz's system.

Lukasiewicz formulated the well known symbolisms. These are written in his work "Elements of Mathematical Logic"(1929). For details of these symbolisms,see the work [1]. He published an axiom system $\mathbb{L}3$ for his Sentential Calculus in 1930.

(1) Axioms of $\mathbb{L}3$:

1 CpCqp.

2 CCpCqrCCpqCpr.

3 CCNpNqCqp.

Cpq : "if p,then q"(implication or conditional).

Np: "it is not true that p"(negation).

(2) Detachment : If α, C$\alpha\beta$,then β (modus ponens).

Definition K: Kpq=NCpNq(conjunction).

$\boxed{1}$ — 2 Rosser's system and Some theses in $\mathbb{L}3$.

(1) Rosser's system.

We are able to have the following theses in $\mathbb{L}3$ by axiomatic method.

① CpKpp. $P \to P \wedge P$.

② CKpqp. $P \wedge Q \to P$.

③ CCpqCNKqrNKrp. $P \to Q. \to . \sim(P \wedge R) \to \sim(R \wedge P)$

Detachment : If α, NKαNβ,then β (modus ponens).

Definition C: Cpq=NKpNq.

These are axioms of Rosser's truth-valued axiom system $\mathbb{R}$(1942)[ ].

$\boxed{2}$ A notation ' $\vdash$ ' and some theorems

We introduce a notations $P_1,P_2,\cdots,P_n \vdash Q$, read as ' $P_1,P_2,\cdots,P_n$ yield Q'.

$\boxed{2}$－1 The notation $P_1,P_2,\cdots,P_n \vdash Q$ indicates that there is a sequence of sentences β_1, $\beta_2,\cdots,\beta_m$ such that β_m is Q and for each β,either:

(1) β_i is an axiom.

(2) β_i is a P.

(3) β_i is the same as some earlier β_j.

(4) β_i derived from two earlier β's by mouds ponens.

More precise version of (4) is: There are j and k,each less than i,such that $\beta P_n \vdash Q$.

$\boxed{2}$－2 General theorems

Theorem 1-1. If $\alpha_1,\alpha_2,\cdots,\alpha_n \vdash \gamma$,then $\alpha_1,\alpha_2,\cdots,\alpha_n,,\beta_1,\beta_2,\cdots,\beta_m \vdash \gamma$.

Proof. Clearly any sequence of δ's which serves for a demonstration of $\alpha_1,\alpha_2,\cdots,\alpha_n \vdash \gamma$ will serve equally well for a demonstration of $\alpha_1,\alpha_2,\cdots,\alpha_n,,\beta_1,\beta_2,\cdots,\beta_m \vdash \gamma$.

Theorem 1-2. If $\alpha_1,\alpha_2,\cdots,\alpha_n \vdash \beta_1$ and $\beta_1,\beta_2,\cdots,\beta_m \vdash \gamma$,then $\alpha_1,\alpha_2,\cdots,\alpha_n,\beta_2,\cdots,\beta_m \vdash \gamma$.

Proof. Let $\delta_1,\cdots,\delta_d$, be a demonstration of $\alpha_1,\alpha_2,\cdots,\alpha_n \vdash \beta_1$ and $\varepsilon_1,\cdots,\varepsilon_e$ be a demonstration of $\beta_1,\beta_2,\cdots,\beta_m \vdash \gamma$. Then each δ is either an axiom or an α or an earlier δ or derived from two earlier δ's by modus ponens,and δ_d is γ. Likewise each ε is either an axiom or a β or an earlier ε or derived from two earlier ε's by modus ponens. If we now construct a new sequence to consist of all the δ's in order followed by all the ε's in order, then we have a demonstration of $\alpha_1,\alpha_2,\cdots,\alpha_n,\beta_2,\cdots,\beta_m \vdash \gamma$. To see this,note that the final step is ε_e,which is γ. Those ε's(if any) which are β_1 are new accounted for as being repetions of δ_d. All δ's and all other ε's are accounted for as before.

Theorem 1-3. If $\alpha_1,\cdots,\alpha_n \vdash \beta_1$, $\gamma_1,\cdots,\gamma_m \vdash \beta_2$ and $\beta_1,\cdots,\beta_b \vdash \delta$, then $\alpha_1,\cdots,\alpha_n$, $\gamma_1,\cdots,\gamma_m,\beta_3,\cdots,\beta_b \vdash \delta$.

Proof. From $\alpha_1,\cdots,\alpha_n \vdash \beta_1$ and $\beta_1,\cdots,\beta_b \vdash \delta$ we have $\alpha_1,\cdots,\alpha_n,\beta_2,\cdots,\beta_b \vdash \delta$ by Theorem 1-2. From $\gamma_1,\cdots,\gamma_m \vdash \beta_2$ and $\alpha_1,\alpha_2,\cdots,\alpha_n,\beta_2,\cdots,\beta_b \vdash \delta$ we get

$\alpha_1,\alpha_2,\cdots,\alpha_n,\gamma_1,\cdots,\gamma_m,\beta_3,\cdots,\beta_b \vdash \delta$ by Theorem 1-2.

Theorem 1-4. If $\alpha_1,\cdots,\alpha_n \vdash \beta$ and $\gamma_1,\cdots,\gamma_m \vdash C\beta\delta$, then $\alpha_1,\alpha_2,\cdots,\alpha_n,\gamma_1,\cdots,\gamma_m \vdash \delta$.

Proof. In Theorem 1-3, take β_1 to be β and β_2 to be $C\beta\delta$, and use Theorem 1-4.

Theorem 1-5. If $\vdash \beta_1$ and $\beta_1,\beta_2,\cdots,\beta_m \vdash \gamma$, then $\beta_2,\cdots,\beta_m \vdash \gamma$.

Corollary. If $\vdash \beta_1$, and $\beta_1,\beta_2,\cdots,\beta_m \vdash \gamma$, then $\beta_2,\cdots,\beta_m \vdash \gamma$.

Proof. This is the special case of Theorem 1-2, which is particularly useful.

Theorem 1-6. If $\vdash \beta_1$, $\vdash \beta_2,\cdots,$ $\vdash \beta_m$, and $\beta_1,\beta_2,\cdots,\beta_m \vdash \gamma$, then $\vdash \gamma$.

Proof. By applying Theorem 1-5 successively m times, we have this.

Theorem 1-7. If $\gamma_1,\cdots,\gamma_m$ is a demonstration of $\alpha_1,\cdots,\alpha_n \vdash \beta$, then for $1 \leqq j \leqq m$, $\gamma_1,\cdots,\gamma_j$ is a demonstration of $\alpha_1,\cdots,\alpha_n \vdash \gamma_j$.

Proof. The j-th step of the demonstration of $\alpha_1,\cdots,\alpha_n \vdash \beta$ is γ_j.

Theorem 1-8. If $\gamma_1,\cdots,\gamma_n$ is any permutation of $\alpha_1,\cdots,\alpha_n$ and if $\alpha_1,\cdots,\alpha_n \vdash \beta$, then $\gamma_1,\cdots,\gamma_n \vdash \beta$.

Proof. It is obvious from the meaning of the demonstration.

Theorem 1-9. If $\alpha_1,\cdots,\alpha_n \vdash \beta$, then $\alpha_1,\alpha_1,\cdots,\alpha_1,\alpha_2,\cdots,\alpha_n \vdash\vdash \beta$ and the converse is valid.

Proof. In theorem 1-1, changing all β's to α_1, we get Theorem 1-9.

$\boxed{2}-3$ Useful theorems.

Theorem 2-1. p, Cpq $\vdash$ q.

Proof.

α_1:p premise
α_2:Cpq premise
α_3:q modus ponens(α_1,α_2).

Theorem 2-2 Cpq, Cqr, p $\vdash$ r.

Proof.

α_1:p premise
α_2:Cpq premise
α_3:q modus ponens(α_1,α_2).
α_4:Cqr premise

Theorem 2-3 CpCqr, q, p $\vdash$ r.

Proof.

α_1:p premise

α_2:CpCqr premise

α_3:Cqr modus ponens(α_1,α_2).

α_4:q premise

α_5:r modus ponens(α_3,α_4)

Theorem 2-4 CpCqr, Cpq, p $\vdash$ r.

Proof.

α_1:p premise

α_2:Cpq premise

α_3: q modus ponens(α_1,α_2)

α_4:CpCqr premise

α_5:r modus ponens(α_3,α_4)

Theorem 2-5 CpCpq, p $\vdash$ q.

Proof.

α_1:p premise

α_2:CpCpq premise

α_3:Cpq modus ponens(α_1,α_2)

α_5:q modus ponens(α_2,α_3)

3. Notations and a conjunctive deduction theorem

$\boxed{3}$— 1 First we shall introduce the following notations:

$K^1p_2=Kp_1p_2$, $K^2p_3=KKp_1p_2p_3$, .

that is, $K^1p_2=Kp_1p_2$, $K^np_{n+1}=KK^{n-1}p_np_{n+1}$, especially $K^0p_1=p_1$.

(1). $CK\cdots KK^up_{u+1}K^vq_{v+1}\cdots K^wr_{w+1}K^{n+2}p_{n+3}$.

where n=u+v+⋯+w and the sequence $(p_1,\cdots,p_{u+1},q_1,\cdots,q_{v+1},\cdots,r_1,\cdots,r_{w+1})$ is any permutation chosen in arbitrary order from the original permutation $(p_1,\cdots,p_{n+3})$, and (n+2)s of K have to be arranged correctly,where $K\cdots KK^up_{u+1}K^vq_{v+1}\cdots K^wr_{w+1}$: antecedent and $K^{n+2}p_{n+3}$: consequent .

(2) $CK^{n+2}p_{n+3}K\cdots KK^up_{u+1}K^vq_{v+1}\cdots K^wr_{w+1}$.

(3) $CK^{n-1}p_np_m$, $(1\leqq m\leqq n)$.

(3) is a special case of 2. Indeed,let p_1=p, p_2=q , p_3=r be written, then $CK^2p_3p_3$= CKKpqrr, $CK^2p_3p_2$=CKKpqrq. $CK^2p_3p_1$=CKKpqrp.

$\boxed{3}$ − 2 A conjunctive deduction theorem.

We have shown the following theorem from Herbrand's Deduction Theorem.[ ]

Theorem 1. If $K^n p_{n+1} \vdash r$, then $K^{n-1} p_n \vdash Cp_{n+1}r$.

We use often this theorem.

Applying the deduction theorem to Theorem 2-1:p, Cpq $\vdash$ q twice time, we have the following two theses:

5 CpCCpqq.

6 CCpqCpq.

Theorem 3-2 Cpq, Cqr $\vdash$ Cpr.

Proof. Applying the deduction theorem to Theorem 2-2:Cpq, Cqr, p $\vdash$ r, we get this.

Applying the deduction theorem to Theorem 3-2 twice, we have the following two theses:

7 CCpqCCqrCpr.

8 CCqrCCpqCpr.

Theorem 3-3 CpCqr $\vdash$ CqCpr.

Proof. Applying the deduction theorem to Theorem 2-3: CpCqr, q, p $\vdash$ r twice, we have this.

Applying the deduction theorem to Theorem 3-3, we have the following thesis:

9 CCpCqrCqCpr.

Theorem 3-2 Cpq, CpCqr $\vdash$ Cpr.

Proof. Theorem 2-3 p/CpCqr, q/Cpq, r/Cpr ,then we get:$\beta_1, \beta_2, \beta_3 \vdash \gamma$ β_1: CCpCqrCCpqCpr , β_2:Cpq , β_3:CpCqr $\vdash$ γ:Cpr As $\vdash \beta_1$ by axiom 1,from Theorem 1-5(m=3) we have this.

Applying the deduction theorem to Theorem 3-2, we get the following thesis:

10 CCpqCCpCqrCpr.

Theorem 3-4. Np, p $\vdash$ q.

Proof. Make a sequence:$\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6 \vdash \gamma$:$\beta_1$:CNpCNqNp, β_2:CCNqNpCpq,β_3:Np, β_4:CNqNp, β_5:Cpq, β_6: p, γ: q. By axiom 1 p/Np, q/Nq, and by axiom 3 p/q, q/p , we have $\vdash \beta_1$ and $\vdash \beta_2$. By modus ponens we derive β_4(β_1,β_3) ,β_5 (β_2,β_4) and γ(β_5,β_6). From Theorem 1-5 (m=6, successively twice) we get this.

Applying the deduction theorem to Theorem 3-4 twice,we have the following two:

11 CNpCpq.

12 CpCNpq.

Theorem 3-5. CNqNp, Cqr, p $\vdash$ r.

Proof. Make a sequence :$\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6 \vdash \gamma$: β_1:CCNqNpCpq, β_2:CNqNp, β_3:Cpq, β_4:p, β_5:q, β_6:Cqr, γ: r. By axiom 3 p/q, q/p, we have $\vdash \beta_1$. By modusponens we derive $\beta_3(\beta_1, \beta_2)$,β_5 (β_3, β_4) and $\gamma(\beta_5, \beta_6)$. From Theorem 1-5 (m=6) we get this.

Applying the deduction theorem 3 times to Theorem 3-5, we get the following thesis:

13 CCNqNpCCqrCpr.

Teorem 3-6. CCpqr $\vdash$ CNpr.

Proof. By Teorem 3-2 p/Np, q/Cpq we have: β_1 :CNpCpq, β_2:CCpqr $\vdash \gamma$:CNpr. By thesis 11 we have $\vdash \beta_1$. From Theorem 1-5 (m=2) we get this.

Applying the deduction theorem to Theorem 3-6 , we get the following thesis:

14 CCCpqrCNpr.

Teorem 3-7. CCNpqr $\vdash$ Cpr.

Proof. By Teorem 3-2, q/CNpq we have: β_1 :CpCNpq, β_2:CCNpqr • γ:Cpr. By thesis 12 we have $\vdash \beta_1$. From Theorem 1-5 (m=2) we get this.

Applying the deduction theorem to Theorem 3-7 , we get the following thesis:

15 CCCNpqrCpr.

Theorem 3-8 CNqq, Nq $\vdash$ p.

Proof. By Theorem 2-4:CpCqr, Cpq, p $\vdash$ r, p/Nq, r/p we have: β_1:CNqCqp, β_2:CNqNq, β_3:Nq • γ:p. As $\vdash \beta_1$ by thesis 11 p/q, q/p, from Theorem 1-5 (m=3) we get this.

Applying the deduction theorem to Theorem 3-8 twice, we have the following thesis:

16 CCNqqCNqp.

Theorem 3-9 CNqp $\vdash$ CCqrCpr.

Proof. We make a sequence β_1:CCNqNpCpq, β_2: CNqNp, β_3:Cpq, β_4:CCpqCCqrCpr $\vdash \gamma$:CCqrCpr. By thesis 11 p/q, q/p, and thesis 10 we get $\vdash \beta_1$. $\vdash \beta_4$. By modus ponens we derive $\beta_3(\beta_1,\beta_2)$ and $\gamma(\beta_3,\beta_4)$. From Theorem 1-5 (m=4) we get this.

Applying the deduction theorem to Theorem 3-9 , we have the following thesis:

17 CCNqpCCqrCpr

Theorem 3-10 CNqq, Cqr, p $\vdash$ r.

Proof. We make a sequence β_1:CNqNq, β_2: CNqNp , β_3:CCqrCpr , β_4: Cqr, β_5:Cpr , β_6:p, $\gamma=\beta_7$: r. By modus ponens, we have $\beta_2(\beta_1$, thesis 16), β_3(thesis17, β_2), β_5 $(\beta_3,\beta_4)$and $\gamma=\beta_7$ (β_5,β_6). Using Theorem 1-2 (n=1,m=2) twice,then we get this.

Applying the deduction theorem to Theorem 3-10 , we have the following thesis:

18 CCNqqCCqrCpr.

Theorem 3-11 CNpp $\vdash$ p.

Proof. By Theorem 3-10 q/p, p/Cpp, r/p we have β_1:CNpp, β_2: Cpp, β_3:Cpp, $\beta_4=\gamma$:p . Clearly $\vdash \beta_2$, $\vdash \beta_3$. From Theorem 1-5 we get this.

Applying the deduction theorem to Theorem 3-11, we have the following thesis:

19 CCNppp.

Theses 7, 12 and 19 are axioms of $\mathbb{L}_{1924}$-system.

Applying the deduction theorem to Theorem 2-5:CpCpq, p $\vdash$ q twice, we get the following thesis:

20 CCpCpqCpq.

$\boxed{3}$ -4 An Enlargement of Theorems.

Theorem 4-1 NNp $\vdash$ CNNpp.

Proof. Make a sequence $\beta_1,\beta_2,\beta_3 \vdash \gamma$. β_1:NNp, β_2:CNpNNNp, β_3:CCNpNNNpCNNpp, γ:CNNpp, where we have β_2 by Theorem 3-4 p/Np ,q/NNNp and β_1 , and β_3 by axiom 3 q/NNp,and by modus ponens(β_2,β_3) we get γ.

Applying the deduction theorem to Theorem 4-1,we have the following thesis:

21 CNNpCNNpp.

Theorem 4-2 NNp $\vdash$ p.

Proof. From Theorem 2-4 p/NNp, q/NNp, r/p we have β_1:CNNpCNNpp, β_2: CNNpNNp, β_3:NNpp, γ:p. By thesis 21 and thesis 5, $\vdash \beta_1$ and $\vdash \beta_2$ respectively. Hence we get this.

Applying the deduction theorem to Theorem 4-2 , we have the following thesis:

22 CNNpp.

We use proof line. Axiom 3 p/NNp, q/Np * C22 p/Np—23.

23 CpNNp.

Theorem 4-3 Cpq $\vdash$ CNqNp.

Proof. $\vdash$ β_1:CNNpp (thesis 22), β_2: Cpq, β_3: CNNpq (MPβ_1, β_2), $\vdash$ β_4:CqNNq (thesis 23 p/q), β_5: CNNppNNq (MPβ_3, β_4), $\beta_6=\gamma$:CNqNp(axiom 3 p/Np, q/Nq). Hence we get this.

Applying the deduction theorem to Theorem 4-3, we have the following thesis:

24 CCpqCNqNp.

Theorem 4-4 Cpq $\vdash$ CNNCqNrNNCrNp.

Proof. From Theorem 3-2 r/Nr ,we have β_1: Cpq, CqNr $\vdash$ CpNr. By Theorem 4-3 q/Nr we get β_2:CpNr $\vdash$ CCrNp. β_3: Cpq, CqNr $\vdash$ CrNp (HS,β_1, β_2), β_4: Cpq $\vdash$ CCqNrCCrNp (β_3 and the deduction theorem) ,β_5: CCqNrCrNp $\vdash$CNCrNp NCqNr(Theorem 4-3 p/CqNr, q/CrNp),$\beta_6=\gamma$:CNCrNpNCqNr $\vdash$ CNNCqNrNNCrNp (Theorem 4-3 p/NCrNp, q/NCqNr). Hence from Theorem 1-5 we get this.

Applying the deduction theorem to Theorem 4-4, we have the following thesis:

25 CCpqCNNCqNrNNCrNp.

Applying Definition: Kpq=NCpNq to Thesis 25, we have the following thesis:

26 CCpqCNKqrNKrp.

This is Axiom 3 of Rosser (1942).

Theorem 4-5 CNpq $\vdash$ CNqp.

Proof. $\vdash$ β_1:CNpq , β_2: CqNNq (thesis 23 p/q), β_3: CNpNNq (Theorem 3-2, β_1, β_2), $\vdash$ β_4:CCNpNNqCNqp (axiom 3, q/Nq), $\beta_5=\gamma$: CNqp (MPβ_3, β_4), Hence from Theorem 1-5 we get this.

Applying the deduction theorem to Theorem 4-5, we have the following thesis:

27 CCNpqCNqp.

Theorem 4-6 CpNq $\vdash$ CqNp.

Proof. $\vdash$ β_1:CpNq , $\vdash$ β_2: CCpNqCNNqNp (thesis 24 q/Nq), β_3: CNNqNp (MP, β_1, β_2), $\vdash$ β_4: CqNNq (thesis 23, p/q), $\beta_5=\gamma$: CqNp (HS, β_3, β_4), Hence from Theorem 1-5 we get this.

Applying the deduction theorem to Theorem 4-6, we have the following thesis:

28 CCpNqCqNp.

$\vdash$ β_1:CNpCpNq (thesis 11 q/Nq), $\vdash$ β_2: CCNpCpNqCNCpNqNNp (thesis 24 p/Np, q/CpNq), β_3: CNCpNqNNp (HS, β_1, β_2), $\vdash$ β_4:CNNpp (thesis 22), $\beta_5=\gamma$: CNCpNqp (HS,β_3, β_4), Hence from Theorem 1-5 we have the following thesis:

29 CNCpNqp.

Applying Definition: Kpq=NCpNq to Thesis 29, we have the following thesis:

30 CKpqp.

This is Axiom 2 of Rosser (1942)

31 CCpCqNrCpCrNq.

$\vdash$ β_1:CCCqNrCrNqCCpCqNrCpCrNq.(thesis 8 q/CqNr, r/CrNq), $\vdash$ β_2: CCqNrCrNq (thesis 28 p/q, q/r), $\beta_3=\gamma$:CCpCqNrCpCrNq (MP , β_1, β_2). Hence we get this.

32 CpNCpNp.

Proof. $\vdash$ β_1:CCpCCpNpNpCpCpNCpNp.(thesis 31 q/CpNp, r/p), β_2: CpCCpNpNp (thesis 5, q/Np), β_3:CpCpNCpNp (MP, β_1,β_2), $\vdash$ β_4:CCpNCpNpCpNCpNp (thesis 20, q/NCpNp), $\beta_5=\gamma$: CpNCpNp (MP,β_3, β_4). Hence we get this.

Applying Definition: Kpq=NCpNq to Thesis 32, we have the following thesis:

33 CpCKpp.

This is Axiom 1 of Rosser (1942). Therefore the set of 33,30 and 26 is the system of the truth-valued axioms by Rosser.

$\boxed{3}$ -5 Theorems and Theses in C-N-K..

Theorem 5-1 Cpq, Cqr $\vdash$ NKNrp.

Proof. Make a sequence β_1,β_2,β_3. β_4,β_5,β_6, $\beta_7=\gamma$. β_1:Cpq, β_2:Cqr=NKqNr (Cpq=NKpNq), β_3:CCpqCNKqNrNKNrp (Thesis 26,r/Nr),β_4:CNKqNrNKNrp (MP,β_1,β_3),$\beta_5=\gamma$:NKNrp(MP,β_2,β_4 $)_6$). Hence we get this.

Applying the deduction theorem to Theorem 5-1,we have the following thesis:

34 CCpqCCqrNKNrp.

Applying Definition: Kpq=NCpNq to Thesis 34, we have the following thesis:

35 NKNpp.

Proof. $\vdash$$\beta_1$:Cpp, $\vdash$ β_2:CCppCCppNKNpp (Thesis 34, q/p. r/p), $\vdash$ β_3 =

γ:NKNpp (MP twice, β_1,β_2). Hence we get this thesis.

36 CNKqrCrNq.

Proof. $\vdash\beta_1$:CCNNqqCNKqrNKrNNq (Thesis 26,p/NNq), $\vdash\beta_2$:CNNqq (Thesis 22, p/q), $\vdash\beta_3$:CNKqrNKrNNq (MP , β_1,β_2), $\vdash\beta_4=\gamma$: CNKqrCrNq(Cpq=NKpNq,β_3). Hence we get this thesis.

Theorem 5-2 Cpq $\vdash$ CKrpKqr.

Proof. β_1:Cpq, β_2:CCpqCNKqrNKrp, β_3:CNKqrNKrp(MP, β_1,β_2), $\vdash\beta_4$: CCNKqrNKrpCKrpKqr (Axiom 3,p/Kqr, q/Krp), $\vdash\beta_5=\gamma$:CKrpKqr.

Applying the deduction theorem to Theorem 5-2,we have the following thesis:

37 CCpqCKrpKqr.

Theorem 5-3 Cpq, Crs $\vdash$ NKNkqsKpr.

Proof. $\vdash\beta_1$:Crs , β_2: Crs $\vdash$ CKprKsp (Theorem 5-2, r/p, p/r, q/s), β_3:CKprKsp (Theorem 1-5,β_1, β_2), β_4:Cpq, β_5:Cpq $\vdash$ CKspKqs (Theorem 5-2, r/s), β_6: CKspKqs (Theorem 1-5, β_4,β_5), β_7:CKprKsp, CKspKqs $\vdash$ NKNkqsKpr (Theorem 5-1, p/Kpr, r/Kqs), β_8:NKNkqsKpr (β_3,β_6,β_7).

38 CKrpKpr.

Proof. $\vdash\beta_1$:Cpp , $\vdash\beta_2$: CCppCKrpKpr (Thesis 27 q/p), $\vdash\beta_3=\gamma$: CKrpKpr(MP, β_1, β_2).

Theorem 5-4 Cpq $\vdash$ CKrpKrq.

Proof. β_1:Cpq , $\vdash\beta_2$: Cpq • CKrpKqr(Theorem 5-2), β_3:CKrpKqr $\vdash$ β_4: CKqrKrq (Thesis 38, r/q, p/r), $\beta_5=\gamma$:CKrpKrq (HS,β_2, β_4). Hence from Theorem 1-5 we get this.

Applying the deduction theorem to Theorem 5-4, we have the following thesis:

39 CCpqCKrpKrq.

Theorem 5-5 Cpq $\vdash$ CKprKqr.

Proof. • β_1:Cpq , $\vdash\beta_2$: CCpqCKrpKqr (Thesis 37), β_3:CKrpKqr (MP, β_1, β_2), $\vdash\beta_4$: CKprKrp (Thesis 38, p/r, r/p), $\beta_5=\gamma$:CKprKqr (HS, β_3, β_4), Hence from Theorem 1-5 we get this.

40 CCpqCKprKqr.

Proof. Applying the deduction theorem to Theorem 5-5, we get this.

41 CNKprNKrp.

Proof. $\vdash\beta_1$:CKrpKpr (Thesis 38), $\vdash\beta_2$: CCKrpKprCNKprNKrp (Thesis 24 p

/Krp, q/Kpr), $\vdash$ β_3:CNKprNKrp (MP, β_1, β_2).

Theorem 5-6 Cpq, Crs $\vdash$ CKprKqs.

Proof. β_1:Cpq , β_2: CKprKqr (Theorem 5-5,β_1), β_3 :Crs, β_4:CKqrKqs,(Teorem 5-4, p/r, q/s, r/q, β_3), $\beta_5=\gamma$:CKprKqs (HS, β_3, β_4).

Theorem 5-7 Cpq, Crs $\vdash$ CpKqr.

Proof. $\vdash$ β_1:Cpq, β_2: Cpr (thesis 28 p/q, q/r), β_3:CKppKqr (Theorem 5-6 r/p, s/r, β_1, β_2), β_4:CpKpp (Theses 33), $\beta_5=\gamma$:CpKqr (HS, β_3, β_4).

Theorem 5-8 KKpqr $\vdash$ r.

Proof. β_1:KKpqr, β_2: CKKpqrKrKpq (Thesis 28 p/r, r/Kpq), β_3:KrKpq (MP, β_1, β_2), β_4:CKrKpqr (Theses 30, p/r, q/Kpq), $\beta_5=\gamma$:r (Mp, β_3, β_4).

Applying the deduction theorem to Theorem 5-8, we have the following thesis:

42 CKKpqrr.

Theorem 5-9 KKpqr $\vdash$ q.

Proof. β_1:KKpqr, β_2: CKKpqrKpq (Thesis 30 p/Kpq, q/r), β_3:Kpq (MP, β_1, β_2), β_4:CKpqKqp (Theses 38, p/q, r/p), β_5:Kqp , β_6:CKqpq (Thesis 30, p/q,q/p) $\beta_7=\gamma$:q (Mp, β_5, β_6).

Applying the deduction theorem to Theorem 5-9, we have the following thesis:

43 CKKpqrq.

Theorem 5-10 KKpqr $\vdash$ p.

Proof. β_1:KKpqr, β_2: CKKpqrKpq (Thesis 30 p/Kpq, q/r), β_3:Kpq (MP, β_1, β_2), β_4:CKpqp (Thesis 30),$\beta_5=\gamma$:p (Mp, β_3, β_4).

Applying the deduction theorem to Theorem 5-9, we have the following thesis:

44 CKKpqrp.

45 CKKpqrKpKqr.

Proof. By Theses 44,43,42, β_1:CKKpqrp, β_2: CKKpqrq , β_3:KKpqrr, β_4: CKKpqrKqr (Theorem 5-7, p/KKpqr, β_2, β_3), β_5:CKKpqrKpKqr (Theorem 5-7, p/KKpqr, q/p,r/Kqr,β_1, β_4).

Theorem 5-11 KpKqr $\vdash$ KKpqr.

Proof. β_1:KpKqr, β_2: CKpKqrKKqrp (Thesis 38 r/p, p/Kqr), β_3:KKqrp (MP, β_1, β_2), β_4: CKKqrpKqKrp (Thesis 42, p/q,q/r,r/p), β_5:KqKrp (Mp, β_3, β_4), β_6:CKqKrpKKrpq (Thesis 38 r/p, p/Kqr), β_7:KKrpq (MP, β_5, β_6),β_8: CKKrpqKrKpq (Thesis 42, p/r,q/p,r/q), β_9:KrKpq (Mp, β_7, β_8) (Thesis 38 r/q, p/Krp),β

$_{10}$:CKrKpqKKpqr (Thesis 38 p/Kpq), β_{11}:KKpqr (Mp, β_{9}, β_{10}).

Applying the deduction theorem to Theorem 5-11, we have the following thesis:

46 CKpKqrKKpqr.

Theorem 5-12 Cpq, CNpq $\vdash$ q.

Proof. β_{1}:Cpq, β_{2}: CNpq, $\vdash$ β_{3}:CCpqCNqNp (Thesis 24), β_{4}:CNqNp (MP, β_{1}, β_{3}), β_{5}:CCNpqCNqNNp (Thesis 24),β_{6}:CNqNNp (Mp, β_{2}, β_{5}), β_{7}: CNqNp,CNqNNp $\vdash$ CNqKNpNNp (Theorem 5-7, p/Nq,q/Np, r/NNp),β_{8}: CNqKNpNNp (β_{7},β_{4},β_{6}), β_{9}:CNKNpNNpNNq (MP, Thesis 24 p/Nq, q/KNpNNp, Mp, β_{8}), β_{10}:CCNpNpNNq (Cpq=NKpNq, p/Np,q/NNp), $\vdash$ β_{11}:CNpNp (Thesis 4,p/Np), β_{12}:NNq (Mp, β_{10}, β_{11}),β_{12}: q (MP,Thesis 22, β_{12}).

Theorem 5-13 CpNq $\vdash$ CpNKqr.

Proof. β_{1}:CpNq, $\vdash$ β_{2}: CKqrq (Thesis 30, p/q, q/r), β_{3}:CCKqrqCNqNKqr (Thesis 24, p/Kqr,), β_{4}: CNqNKqr (MP, β_{2}, β_{3}), β_{5}:CpNKqr (HS,β_{1}, β_{3}).

Corollary. CpNq $\vdash$ CpCqr.

Proof. β_{1}:CpNq, β_{2}: CpNq $\vdash$ CpNKqNr (Theorem 5-13, r/Nr),β_{3}:CpNKqNr (Theorem 1-2, β_{1}, β_{2}), β_{4}:CpCqr (Cpq=NKpNq, p/q,q/r, β_{3}).

Applying the deduction theorem to Theorem 5-13, we have the following thesis:

47 CCpNqCpNKqr.

Applying the deduction theorem to Corollary to Theorem 5-13 , we have the following thesis:

48 CCpNqCpCqr.

Theorem 5-14 Cpr $\vdash$ CpCqr.

Proof. β_{1}:Cpr, $\vdash$ β_{2}: CCprCqCpr (Axiom 1 , p/Cpr,), $\vdash$ β_{3}:CqCpr (MP,β_{1}, β_{2}), $\vdash$ β_{4}: CCqCprCpCqr (Thesis 9,p/q,q/p), β_{5}:CpCqr (MP, β_{3}, β_{4}).

Applying the deduction theorem to Theorem 5-14 , we have the following thesis:

49 CCprCpCqr.

50 CCKpqrCpCqr.

Proof. $\vdash$ β_{1}:CCKpKqNrKKpqNr (Thesis 46, r/Nr), β_{2}:CNNKqNrKqNr (Thesis 22,p/KqNr), $\vdash$ β_{3}:CKpNNqNrKpKqNr (MP, Theorem 5-4, p/NNKqNr, q/KqNr, r/p, β_{2}), $\vdash$ β_{4}:CKpNCqrKpKqNr (Cpq=NKpNq, q/Cqr,β_{3}), $\vdash$ β_{5}:CKpNCqrKKpqNr (HS,β_{1}, β_{4}), $\vdash$ β_{6}:CNKKpqNrNKpNCqr (Thesis 24, p/KpNCqr, q/KKpqNr, β_{5}), $\beta_{7}=\gamma$:CCKpqrCpCqr (Cpq=NKpNq).

51 CpCqKpq.

Proof. $\vdash$ β_1:CCKpqKpqCpCqKpq (Thesis 50, r/Kpq), $\vdash$ β_2:CkpqKpq (Thesis 4, p/Kpq), $\vdash$ β_3:CpCqKpq (MP, β_1, β_2).

Theorem 5-15 Cpq $\vdash$ CpNNq.

Proof. $\vdash$ β_1:Cpq, (Thesis 23,p/q), $\vdash$ β_2:CqNNq (HS, β_1, β_2).

Applying the deduction theorem to Theorem 5-15 , we have the following thesis:

52 CCpqCpNNq.

Theorem 5-16 Cpq,Crs $\vdash$ NKNKqsKpr.

Proof. β_1:Crs, β_2: CCrsCKprKsp (Theorem5-2,p/r,q/s,r/p), $\vdash$ β_3:CKprKsp (MP, β_1, β_2), β_4: Cpq , β_5:CCpqCKspKqs (Theorem 5-2, r/s), β_6 : CKspKqs(MP, β_3, β_4), $\vdash$ β_7:CKprKsp, CKsKqs $\vdash$ NKNKqsKpr (Theorem 5-1, p/Kpr, q/Ksp, r/Kqs), β_8: NKNKqsKpr (β_3, β_6, β_7).

53 CKrNNpKpr.

Proof. $\vdash$ β_1:CNNpp $\vdash$ CKrNNpKpr(Theorem5-2,p/NNp,q/p), $\vdash$ β_2:CNNpp (Thesis 22,p/r,q/s,r/p), $\vdash$ β_3:CKrNNpKpr (MP, β_1, β_2).

54 CKrNNpKrp.

Proof. $\vdash$ β_1:CKrNNpKpr(Thesis 52), $\vdash$ β_2:CKprKrp (Thesis 38, p/r,,r/p), $\vdash$ β_3: CKrNNpKrp (MP, β_1, β_2).

Theorem 5-17 CKpqr, CKpNqr $\vdash$ Cpr.

Proof. β_1:CKpqr, β_2:CKpNqr , $\vdash$ β_3:CKqpKpq (Thesis 38,r/q), β_4:CKqpr(HS, β_1, β_3), $\vdash$ β_5:CKNqpKpNq (Thesis 38, r/Nq), β_6 : CKNqpr(HS, β_2, β_5), $\vdash$ β_7: CKqprCqCpr (Thesis 50, p/q,q/p), β_8: CqCpr(MP, β_4, β_7), $\vdash$ β_9: CKNqprCNqCpr (Thesis 50, p/Nq,q/p), β_{10}: CNqCpr (MP, β_6, β_9), $\vdash$ β_{11}:CqCpr ,CNqCpr $\vdash$ Cpr (Teorem 5-2, p/q, q/Cpr), β_{12}:Cpr (β_8, β_{10}, β_{11}).

Theorem 5-18 CpNr $\vdash$ CpNKqr.

Proof. β_1:CpNr, $\vdash$ β_2:CpNr $\vdash$ CpNKrq (Theorem 5-13,q/r,r/q), β_3:CpNKrq (β_1, β_2), $\vdash$ β_4:CNKrqNKqr(Thesis 41,p/r,r/q), β_5:CpNKqr (HS, β_3, β_4).

Applying the deduction theorem to Theorem 5-18 , we have the following thesis:

55 CCpNrCpNKqr.

Theorem 5-19 Cpq, CpNr $\vdash$ CpNCqr.

Proof. β_1:Cpq, β_2:CpNr. By Theorem 5-7 r/Nr, β_1 and β_2, we get β_3:CpKqNr. By Theorem 5-15 q/KqNr and β_3,we get β_4:CpNNKqNr , β_5 : CpNCqr(Cpq=NKpNq, p/q,q/p).

END

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

A Real-Time Specification Language

Fernando Naufel do Amaral, Edward Hermann Haeusler, Markus Endler
Dept. of Informatics, PUC-RJ, Brazil
{fnaufel, hermann, endler}@inf.puc-rio.br

Abstract. A specification language for real-time software systems is presented. Notions from Category Theory are used to specify how the components of a system should interact. The potential role of the proposed language in the search for interoperability of specification formalisms is briefly discussed.

1 Introduction

The aim of this work is to present RT-Community, a specification language for real-time reactive systems. Using the language, one is able to specify the computations of the individual components of a real-time system as well as the way these components interact with each other. This makes RT-Community suitable as an architecture description language (ADL) in the sense of [1].

RT-Community is an extension of the specification language Community ([7]), from which it inherits its characteristics as a coordination language ([4]). Roughly speaking, this means that it supports the separation between computational concerns (what each component does) and coordination concerns (how components are put together and how they interact to present the behavior expected of the system as a whole).

This paper presents the syntax and informal semantics of RT-Community in section 2 (a formal model-theoric semantics, not included here for lack of space, is found in [2]). Section 3 shows how composition is done in RT-Community by way of some basic notions of Category Theory. Finally, section 4 offers some concluding remarks, especially about the potential role of RT-Community in the search for interoperability of specification formalisms.

2 RT-Community

An RT-Community component has the form shown in Figure 1, where

- V is a finite set of variables, partitioned into input variables, output variables and private variables. Input variables are read by the component from its environment; they cannot be modified by the component. Output variables can be modified by the component and read by the environment; they cannot be modified by the environment. Private variables can be modified by the component and cannot be seen (i.e. neither read nor modified) by the environment. We write $loc(V)$ for $prv(V) \cup out(V)$. We assume given but do not make explicit the specification of the data types over which the variables in V range.

component P
in in (V)
out out (V)
prv prv (V)
clocks C
init F
do
$\Box_{g \in sh(\Gamma)}$ $g : [T(g), B(g) \rightarrow R(g), \ \|_{v \in D(g)} \ v := F(g,v)]$
$\Box_{g \in prv(\Gamma)}$ **prv** $g : [T(g), B(g) \rightarrow R(g), \ \|_{v \in D(g)} \ v := F(g,v)]$
end component

Figure 1: The form of an RT-Community component

- C is a finite set of clocks. Clocks are like private variables in that they cannot be seen by the environment; they can be consulted (i.e. their current values may be read) by the component; however, the only way a component can modify a clock variable is by resetting it to zero.
- F is a formula specifying the initial state of the component (i.e., conditions on the initial values of variables in $loc(V)$). The initial values of input variables are unconstrained, and clocks are always initialized to zero.
- Γ is a finite set of action names. Actions may be either shared or private. Shared actions are available for synchronization with actions of other components, whereas the execution of private actions is entirely under control of the component.
- The body of the component consists of a set of instructions resembling guarded commands. For each action $g \in \Gamma$, we have the *time guard* $T(g)$, which is a boolean expression over atomic formulae of the form $x \sim n$ and $x - y \sim n$, where $x, y \in C, n \in \mathbb{R}_{\geq 0}$ and $\sim$ is one of $\{<, >, =\}$; the *data guard* $B(g)$, which is a boolean expression constructed from the operations and predicates present in the specification of the data types of V; the *reset clocks* $R(g) \subseteq C$, containing the clocks to be reset upon execution of g; and the *parallel assignment* of a term $F(g, v)$ to each variable v that can be modified by g. We denote by $D(g)$ the set of variables that can be modified by g. This set will be called the write frame (or domain) of g.

The execution of a component proceeds as follows: at each step, an action whose time and data guards are true (such an action is said to be *enabled*) may be executed. If more than one enabled action exists, one may be selected non-deterministically. If an action is selected, the corresponding assignments are effected and the specified clocks are reset. If no action is selected, time passes, the component idles and all clocks are updated accordingly (i.e., the semantics is based on global time). However, three conditions must be met: (1) a private action may not be enabled infinitely often without being infinitely often selected (the fairness condition), (2) time may not pass if there is an enabled private action (the urgency condition), and (3) a shared action may not be disabled solely by the passage of time (the persistency condition).

A model-theoretic semantics of RT-Community based on Time-Labelled Transition Systems can be found in [2].

```
component snooze
    in        float initialInterval, float minimum
    out       bool ringing
    prv       float interval
    clocks    c
    init      $\neg ringing \wedge interval = -1$
    do
    [] firstRing:       $\rightarrow ringing := \mathbf{true} \parallel interval := initialInterval$
    [] snooze:          $ringing \wedge interval > minimum \rightarrow$
                            $\mathbf{reset}(c) \parallel ringing := \mathbf{false} \parallel interval := interval/2$
    [] off:             $ringing \rightarrow ringing := \mathbf{false} \parallel interval := -1$
    [] prv timeout:     $c == interval \rightarrow ringing := \mathbf{true}$
end component
```

Figure 2: The *snooze* component

2.1 A Simple Example

We present a component for the "snooze" feature of an alarm clock. Component *snooze* is activated when the timekeeping component of the alarm clock (not shown) reaches the preset time, as indicated by action *firstRing*. This action sets the output variable *ringing* to true, a change that may be detected by a "bell" component (not shown either). If the user presses the "off" button at this point, the alarm and the snooze component are turned off, as indicated by the *off* action. However, if the user presses the "snooze" button (action *snooze*), the alarm stops ringing, only to ring again after a preset time interval. This second ringing of the alarm is activated by the *snooze* component upon detecting the timeout (private action *timeout*). Now, if the user presses the "snooze" button this time, he will be allowed to sleep for an additional period with half the duration of the initial interval. This pattern repeats, with the interval being halved each time the alarm rings and the user presses the "snooze" button, until either the user presses the "off" button or the interval reaches a certain minimum duration (in this last case, the alarm will go on ringing until the user presses the "off" button).

The duration of the initial interval and the minimum duration are provided by the environment of the *snooze* component, as indicated by input variables *initialInterval* and *minimum*. The specification of the *snooze* component is given in Figure 2. There, time guards and data guards that have the constant truth value **true** are omitted, and the resetting of a clock c is indicated by the **reset**(c) instruction.

3 Composing RT-Community Components

We use basic concepts from Category Theory ([6, 10]) to define how composition is done in RT-Community. The use of Category Theory allows us to describe the interaction of components in an abstract fashion, thus establishing the essentials of their coordination in such a way that RT-Community components may be replaced by specifications in other formalisms (e.g. a temporal logic with time-bounded operators such as MTL – see [3]), a desirable feature when our ultimate goal is to promote the interoperability of formalisms.

3.1 The Categories of Signatures and Components

Formally, an RT-Community component is a pair (Σ, Δ) with Σ the signature and Δ the body of the component.

Definition 3.1 (Signature of a component). *An RT-Community signature is a tuple* $\Sigma = < V, C, \Gamma, tv, ta, D, R >$, *where* V *is a finite set of variables,* C *is a finite set of clocks,* Γ *is a finite set of action names,* $tv : V \longrightarrow \{in, out, prv\}$ *is the typing function for variables,* $ta : \Gamma \longrightarrow \{shr, prv\}$ *is the typing function for action names,* $D : \Gamma \longrightarrow 2^{loc(V)}$ *is the write frame function for action names, and* $R : \Gamma \longrightarrow 2^{C}$ *is the reset clock function for action names.*

Definition 3.2 (Body of a component). *The body of an RT-Community component with signature* Σ *is a tuple* $\Delta = < T, B, F, I >$, *with* $T : \Gamma \longrightarrow \mathcal{PROP}(C)$ *the time guard function for action names,* $B : \Gamma \longrightarrow \mathcal{PROP}(V)$ *the data guard function for action names,* $F : \Gamma \longrightarrow (loc(V) \longrightarrow \mathcal{TERM}(V))$ *the assignment function for action names, and* I *a formula over* $loc(V)$ *specifying the initial state(s) of the component. Here,* $\mathcal{PROP}(C)$ *denotes the set of boolean propositions over clocks,* $\mathcal{PROP}(V)$ *denotes the set of boolean propositions over variables and* $\mathcal{TERM}(V)$ *is the set of terms of the term algebra of the data types involved. Function* F *must respect sorts when assigning terms to variables.*

We define the category of RT-Community signatures. In what follows, it is helpful to keep in mind that Σ_1 is the signature of a component (referred to as "the component") that is embedded in a system (referred to as "the system") whose signature is Σ_2.

Definition 3.3 (Category of signatures). *Sign is the category that has signatures of RT-Community as objects; a morphism* $\sigma : \Sigma_1 \longrightarrow \Sigma_2$ *in Sign is a triple* $< \sigma_v, \sigma_c, \sigma_a >$ *defined as follows:*

$\sigma_v : V_1 \longrightarrow V_2$ is a total function such that

- For all $v \in V_1$, $sort_2(\sigma_v(v)) = sort_1(v)$. Variables of the component are mapped to variables of the system in such a way that sorts are preserved;
- For all $o, i, p \in V_1$, $o \in out(V_1) \Rightarrow \sigma_v(o) \in out(V_2)$, $i \in in(V_1) \Rightarrow \sigma_v(i) \in in(V_2) \cup out(V_2)$, $p \in prv(V_1) \Rightarrow \sigma_v(p) \in out(V_2)$. The nature of each variable is preserved, with the exception that input variables of the component may become output variables of the system. This is because, as will be seen below, an input variable of a component may be "connected" to an output variable of another component; when this happens, the resulting variable must be considered an output variable of the system: its value can be modified by the system, and it remains visible to the environment (which, however, cannot modify it).

$\sigma_c : C_1 \longrightarrow C_2$ is a total, injective function from the clocks of the component to the clocks of the system. In other words, all clocks of the component must retain their identity in the system.

$\sigma_a : \Gamma_2 \longrightarrow \Gamma_1$ is a partial function from the actions of the system to the actions of the component. σ_a is partial because an action of the system may or may not correspond to an action of the component; i.e., if the component does not participate in a given action

g of the system, then $\sigma_a(g)$ is undefined. Furthermore, note the contravariant nature of σ_a compared to σ_v and σ_c: because each action of the system can involve at most one action of the component, and each action of the component may participate in more than one action of the system, the relation must be functional from the system to the component. Besides, σ_a must satisfy the following conditions ($D(v)$ for a variable v denotes the set of actions having v in their domain):

- For all $g \in \Gamma_2$ for which $\sigma_a(g)$ is defined, $g \in shr(\Gamma_2) \Rightarrow \sigma_a(g) \in shr(\Gamma_1)$ and $g \in prv(\Gamma_2) \Rightarrow \sigma_a(g) \in prv(\Gamma_1)$. An action of the system is of the same nature (shared or private) as the component action involved in it.
- For all $g \in \Gamma_2$ for which $\sigma_a(g)$ is defined, and for all $v \in loc(V_1)$, $v \in D_1(\sigma_a(g)) \Rightarrow \sigma_v(v) \in D_2(g)$ and $g \in D_2(\sigma_v(v)) \Rightarrow \sigma_a(g) \in D_1(v)$. If a component variable is modified by a component action, then the corresponding system variable is modified by the corresponding system action. Besides, system actions where the component does not participate cannot modify local variables of the component.

  The following item states analogous conditions for clocks of the component:
- For all $g \in \Gamma_2$ for which $\sigma_a(g)$ is defined, and for all $c \in C_1$, $c \in R_1(\sigma_a(g)) \Rightarrow \sigma_c(c) \in R_2(g)$ and $g \in R_2(\sigma_c(c)) \Rightarrow \sigma_a(g) \in R_1(c)$.

We define the category of RT-Community components, with whole components as objects and special signature morphisms between them.

Definition 3.4 (Category of components). *Comp is the category that has components of RT-Community as objects; a morphism $\sigma : (\Sigma_1, \Delta_1) \longrightarrow (\Sigma_2, \Delta_2)$ in Comp is a signature morphism $\sigma : \Sigma_1 \longrightarrow \Sigma_2$ satisfying the following conditions:*

- For all actions g in Γ_2 with $\sigma_a(g)$ defined, we have $\Phi \models B_2(g) \rightarrow \bar{\sigma}(B_1(\sigma_a(g)))$ and $\Phi \models T_2(g) \rightarrow \bar{\sigma}(T_1(\sigma_a(g)))$, where Φ is a suitable axiomatization of the specification of the data types involved, and $\bar{\sigma}$ is the extension of σ_v to the language of terms and propositions. The behavior of the system (Σ_2, Δ_2) is such that an action g of the system cannot disrespect the data and time guards of the corresponding action $\sigma_a(g)$ of the component; i.e., the system can only strengthen said guards.
- $\Phi \models I_2 \rightarrow \bar{\sigma}(I_1)$, with Φ and $\bar{\sigma}$ as above. The initial state of the component must be implied by the initial state of the system.
- For all actions g in Γ_2 with $\sigma_a(g)$ defined, and for all local variables v in $D_1(\sigma_a(g))$, we have $F_2(g)(\sigma_v(v)) = \bar{\sigma}(F_1(\sigma_a(g))(v))$, where, as before, $\bar{\sigma}$ is the extension of σ_v to the language of terms and propositions. Recall that F is the function that assigns to each action g a mapping from the variables in the action's domain $D(g)$ to terms of the term algebra of the data types involved. This means that an action g of the system can only assign to a local variable of the component the "translation" of the value that the corresponding action $\sigma_a(g)$ of the component does.

3.2 Channels and Configurations

The interaction of two components P_1 and P_2 may be either in the form of the connection of variables or in the form of the synchronization of actions (or both). This interaction is specified using a third component, called a *channel*. Given components P_1 and P_2 and a channel P_c, the composition of P_1 and P_2 via P_c is represented by the diagram

$$\begin{tikzcd} P_1 & P_c \arrow[l, "\sigma_1"'] \arrow[r, "\sigma_2"] & P_2 \end{tikzcd}$$

When certain conditions are met, such a diagram is called a *configuration*. In a configuration, morphisms σ_1 and σ_2 specify how P_1 and P_2 should interact, as discussed below:

Variables: The only variables that the channel P_c can contain are of the input kind, so they can be mapped to input or output variables of P_1 and P_2. Given an input variable v of P_c, we say that variables $\sigma_1(v)$ and $\sigma_2(v)$ of P_1 and P_2, respectively, are connected. If two input variables are connected, the result will be an input variable of the composite system. If an input variable and an output variable are connected, the result will be an output variable of the composite system. We do not allow two output variables to be connected (a diagram where this happens is not considered a well-formed configuration). Furthermore, private variables and clocks cannot be connected, so we do not allow the channel P_c to have private variables or clocks.

Actions: The only actions that the channel P_c can contain are of the shared kind. Furthermore, these actions must be "empty" in the sense that their time and data guards are the constant values **true**, they do not reset any clocks and do not modify any variables (i.e. their write frame is empty). This means that the channel P_c is a "neutral" component with no behavior of its own, serving only as a kind of "wire" for joining the actions of P_1 and P_2.

Given an action g_1 of P_1 and an action g_2 of P_2 having $\sigma_1(g_1) = \sigma_2(g_2) = g$ for g an action of P_c, we say that g_1 and g_2 are synchronized. When two actions are synchronized, the result will be a joint action of the composite system. This joint action will be of the shared kind, its data and time guards being the conjunction of the corresponding guards of g_1 and g_2, its effect being to reset the union of the sets of clocks reset by g_1 and g_2 and to perform the assignments in the union of the sets of assignments dictated by g_1 and g_2.

Formally, the system that results from the composition of P_1 and P_2 through channel P_c is the pushout (in the category $\underline{Comp}$) of the configuration above, written $P_1 +_{P_c} P_2$. In general, for a configuration diagram involving any (finite) number of channels and components, the resulting system is given by the colimit of the diagram. A configuration is said to be well-formed when it satisfies the conditions described above. For well-formed configurations, the colimit will always exist.

3.3 A Simple Example

In order to illustrate composition in RT-Community, we present the *timekeeping* component of the alarm clock discussed in Section 2.1 and show how the *timekeeping* and *snooze* components can be put together. The *snooze* component was shown in Figure 2. The *timekeeping* component is shown in Figure 3. There, it is assumed that a data type *Time* is available, and that adding 1 to a variable of this data type corresponds to adding one second to the time value it contains. Besides, the fact that an action performs no assignments and resets no clocks (like the *ring* action) is indicated by the abbreviation **skip**.

```
component timekeeping
    in          Time alarmTime, Time currentTime
    out         float snoozeInterval, float minimum
    prv         int ticksPerSec, Time now, boolean alarmOn
    clock       c
    init        snoozeInterval = 10 ∧ minimum = 1 ∧ ticksPerSec = ... ∧ ¬ alarmOn
    do
    [] setTime:          → now := currentTime || reset(c)
    [] setAlarm:         → alarmOn := true
    [] ring:             alarmOn ∧ now == alarmTime → skip
    [] alarmOff:         → alarmOn := false
    [] prv keepTime:     c == ticksPerSec → now := now + 1 || reset(c)
end component
```

Figure 3: The *timekeeping* component

When composing the *timekeeping* and *snooze* components, we want to identify variables *snoozeInterval* and *initialInterval*, as well as variables *minimum* (in *timekeeping*) and *minimum* (in *snooze*), so that the input variables in *snooze* will contain the constant values provided by *timekeeping*.

Furthermore, we want to synchronize actions *ring* and *firstRing* so that *snooze* will be activated exactly when *timekeeping* detects that the current time equals the time the alarm has been set to ring. We also want to synchronize the *alarmOff* and *off* actions, meaning that when the user presses the "off" button, both the snooze and the alarm mechanisms are turned off.

Notice that the resulting composite system is still open, in the sense that there are still unconnected input variables: *currentTime* and *alarmTime* receive values given by the user when he or she wants to set the time or the alarm, operations that are made available by the shared actions *setTime* and *setAlarm*, respectively.

Further interaction with the environment is given by the following features:

- The *snooze* action of the *snooze* component remains as a shared action of the system; it must be synchronized with an action of the environment representing the pressing of the "snooze" button while the bell is ringing.
- The joint action (*alarmOff* | *off*) is a shared action of the system that must be synchronized with an action of the environment representing the pressing of the "off" button while the bell is ringing.
- The output variable *ringing* in the *snooze* component must be connected to an input variable of a component representing the actual bell mechanism.

4 Concluding Remarks

RT-Community, like untimed Community, lends itself to the specification of architectural connectors that express interaction of a more complex nature than in the examples presented here. Other features of the language include the capacity for underspecification, such as lower and upper bounds for action data guards (i.e., safety conditions and progress conditions, respectively), and the partial specification of the effect of an action g on its write frame $D(g)$, by means of a boolean expression (involving primed variables, as is customary in other formalisms) instead of parallel assignment.

When specifying the components of a real-time system, it may be appropriate to use formalisms of a higher level of abstraction than RT-Community. One example would be the use of a real-time temporal logic (e.g. MTL – [3]) to specify the behavior of a component. In fact, we expect that by using mappings between logics such as those described in [5], a wider range of formalisms may be employed in the specification of a single system, allowing for a situation of interoperability among logics and specification languages.

The adaptation of RT-Community to systems involving mobility and dynamic reconfiguration is the subject of current study. A similar goal is being pursued in relation to untimed Community ([9]), where graph rewriting is used to reflect runtime changes in the system. We are investigating an alternative approach, where channels may be passed between components to allow them to engage in new connections at execution time – a strategy similar to the one used in π-calculus ([8]).

RT-Community is currently being contemplated for the specification of hypermedia presentations, a domain where real-time constraints occur naturally.

References

[1] R. Allen and D. Garlan, A Formal Basis for Architectural Connectors, ACM TOSEM, 6(3)(1997) 213–249.

[2] F.N. Amaral and E.H. Haeusler, A Real-Time Specification Language, Technical Report, Dept. of Informatics, PUC-RJ, Brazil (2002).

[3] E. Chang, Compositional Verification of Reactive and Real-Time Systems, PhD Thesis, Stanford University (1995).

[4] D. Gelernter and N. Carriero, Coordination Languages and their Significance, Comm. ACM **35**, 2 (1992) 97–107.

[5] A. Martini, U. Wolter and E.H. Haeusler, Reasons and Ways to Cope with a Spectrum of Logics, in J. Abe and J.I. da S. Filho (eds.), Logic, Artificial Intelligence and Robotics (proc. LAPTEC 2001), series: Frontiers in Artificial Intelligence and Applications **71**, IOS Press, Amsterdam (2001) 148–155.

[6] B. Peirce, Basic Category Theory for Computer Scientists, The MIT Press (1991).

[7] J.L. Fiadeiro and A. Lopes, Semantics of Architectural Connectors, in M. Bidoit and M. Dauchet (eds), TAPSOFT'97, LNCS 1214, Springer-Verlag (1997) 505-519.

[8] R. Milner, Communicating and Mobile Systems: the π-Calculus, Cambridge University Press (1999).

[9] M. Wermelinger and J.L. Fiadeiro, Algebraic Software Architecture Reconfiguration, in Software Engineering–ESEC/FSE'99, volume 1687 of LNCS, Springer-Verlag (1999) 393–409.

[10] G. Winskell and M. Nielsen, Categories in Concurrency, in A.M. Pitts and P. Dybjer (eds.), Semantics and Logics of Computation, Cambridge University Press (1997) 299–354.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Defuzzification in Medical Diagnosis

Pereira JCR[1], Tonelli PA[2], Barros LC[3] and Ortega NRS[4]

[1] *School of Public Health, University of São Paulo*
[2] *Institute of Mathematics and Statistics, University of São Paulo*
[3] *Institute of Mathematics, Statistics and Scientific Computation, University of Campinas*
[4] *School of Medicine-LIM01/HCFMUSP, University of São Paulo*
Av. Dr. Arnaldo 455, São Paulo, 01246-903, SP, Brazil

Abstract

In this work, we present a classification of patients through relational fuzzy signs/disease matrixes. This relational fuzzy matrixes was elaborated based on real cases and through two methods: max-min composition and Gödel implication. The results found from the different methods was compared and discussed in the medical point of view. The necessity of the defuzzification process in a medical environment is also discussed.

Keywords: Fuzzy Relations, max-min Composition, Gödel Implication, Medical diagnosis, Defuzzification.

1. Introduction

Fuzzy systems concerning medical diagnosis often seek the allocation of patients to specific nosologic categories through some sort of defuzzification rule. This follows the rationale of medical practice under which for a patient presenting with a given set of clinical signs and symptoms a conclusive diagnosis should be produced.

Such conclusion, nonetheless, will always hinge on premises of previous composite relational equations, which analyzing records of past patients concerning symptoms & signs and diagnoses, establish how they are related. Depending on the model used in this analysis, different sort of information will be available for the definition of a defuzzification procedure. In the present paper, the yields of two different methods are studied.

2. Methods

A data set of 153 children recording their clinical signs and diagnoses was randomly separated into an analysis sample (75% of cases) and a validation sample (25% of cases), the former been used to derive a matrix of relations between diagnoses and signs, and the latter been used to assess performance in patient allocation to a diagnosis category according with the aforementioned matrix. Diagnoses were pneumonia (d_1), diseased but not pneumonia (d_2), and healthy (d_3), crisply assessed as either present or absent. Clinical signs were chest X-ray (s_1), dyspnoea (s_2), chest auscultation (s_3), cardiac rate (s_4), body temperature (s_5), toxaemia (s_6), respiratory rate (s_7), which were originally assessed through qualitative scales, whose values were normalised to the unit to express different grades of membership as a linear relation between absence and maximum presentation.

The relational models compared were a max-min (sup-min) composition and Gödel's implication [1,2,3]. Max-min relation is defined as:

Max-min relation: Given S and T, two binary relations of U × V and V × W (*e.g. signs* × *patient and patient* × *diagnosis*), the composition sup-min (*e.g. signs* × *diagnoses*) is a fuzzy relation in U × W of the type

$$(S * T)(x,z) = \sup_{y \in V}[\min(S(x,y),T(y,z))] \qquad (1)$$

Gödel's implication: Gödel's implication treats Cartesian products in a way that preserves the truth table. The relational equation that incorporates Gödel's implication is given by

$$(S *_g T)(x,z) = \inf_{y \in V}[g(S(x,y),T(y,z))] \qquad (2)$$

and g is defined as a function in [0,1] × [0,1], such that:

$$g : [0,1] \times [0,1] \rightarrow [0,1]$$

where

$$\begin{aligned} g(a,b) &= a \Rightarrow b \\ &= \sup\{x \in [0,1] : \min(a,x) \leq b\} \\ &= \begin{cases} 1 & if \quad a \leq b \\ b & if \quad a > b. \end{cases} \end{aligned}$$

To allocate patients from the validation sample to the diagnosis categories, for each patient (P_n) allocation for diagnoses (d_m) was drawn from the composition of his vector of clinical signs (s_i) with the relational matrix of each model according with the function

$$R(P_n)(d_m) = \sup_{1 \leq i \leq 7}[\min[R(d_m,s_i),P_n(s_i)]] \qquad (3)$$

The defuzzification rule was defined as the maximum points of the resulting membership functions [1], in other words, one patient should be allocated to the diagnosis to which he had highest membership. As the relation between signs and healthy could achieve some degree of membership, a patient should be allocated to the healthy category if his membership to both pneumonia and other disease were null.

To assess performance of defuzzification under each model, the overall agreement between results and the known classification of patients in the validation sample was calculated.

3. Results

Derived from the analysis sample, the matrixes of relations between clinical signs and diagnoses under each model were:

Table 1 - Relationship matrixes for the two models studied.

max-min

	d_1	d_2	d_3
s_1	0.43	0	0
s_2	1	0.25	0
s_3	0.67	0	0
s_4	1	1	0.5
s_5	0.67	1	0
s_6	0.75	0.5	0
s_7	1	0.75	0

Gödel

	d_1	d_2	d_3
s_1	1	0	0
s_2	0	0	0
s_3	1	0	0
s_4	0	0	0
s_5	0	0	0
s_6	0	0	0
s_7	0	0	0

Max-min composition better elicits relations of intersection, or equivalence, or interspersing, and thus even healthy subjects (d_3) can present with some degree of a clinical sign, which resulted in cardiac rate (s_4) having some relation with the healthy diagnosis (actually, 6 children had mild tachycardia, 4 of which had weights below the 25° percentile, thus probably not being perfectly healthy but having malnutrition and the not rarely accompanying anemia, which could account for the tachycardia). Gödel exams implication, thus causal relationship, and in the present study, as a corollary of the diagnoses being crisply assessed, this resulted in selecting signs which were pathognomonic to pneumonia (s_1 and s_3, chest X-ray and chest auscultation).

Applying equation (3) for each model and patient, a vector of values representing membership of each patient to diagnoses 1 to 3, namely pneumonia, diseased not pneumonia, and healthy, was accomplished. This had the form of what is presented below for a sample of patients:

Table 2 - Patient membership to diagnosis category according to each model.

max-min

Patient ID	pneumonia	other disease	healthy
5	0.75	0.5	0.5
10	0.25	0.25	0.25
23	0.75	0.5	0
31	0.75	0.75	0.5
37	0.75	0.75	0.5
38	0.33	0.25	0.25
etc.			

Gödel

Patient ID	pneumonia	other disease	healthy
5	0.43	0	0
10	0.14	0	0
23	0.33	0	0
31	0.67	0	0
37	0.67	0	0
38	0.33	0	0
etc.			

Applying the defined defuzzification rule, in max-min model there were situations when allocation of patients was missed due to ties in membership functions, e.g. patient n° 10 under. An alternative was to allow multiple allocation, so that a patient could be simultaneously allocated to more than one category. Agreement between allocation of patients through the two models information and the patient real status in the validation sample is shown in tables 3 to 5:

Table 3 - Patient allocation according with max-min relation matrix taken as a single response variable.

		patient real status			
		pneumonia	other disease	healthy	Total
max-min	pneumonia	3			3
	other disease				
	healthy		4	11	15
	Total	3	4	11	18
Overall agreement: 77.8% Valid cases: 18 Missing cases: 20					

Table 4 - Patient allocation according with max-min relation matrix taken as a multi response variable.

		patient real status			
		pneumonia	other disease	healthy	Total
max-min	pneumonia	10	9	4	23
	other disease	7	9	4	20
	healthy		4	11	15
	Total	10	13	15	38
Overall agreement: 78.3% Valid cases: 38 Missing cases: 0					

Table 5 - Patient allocation according with Gödel's relation matrix.

		patient real status			
		pneumonia	other disease	Healthy	Total
Gödel	pneumonia	10			10
	other disease				
	healthy		13	15	28
	Total	10	13	15	38
Overall agreement: 65.8% Valid cases: 38					

4. Comments and Conclusions

> *Where is the knowledge we have lost in information, where is the wisdom we have lost in knowledge.*
>
> T.S. Elliot

A defuzzification procedure may be inescapable in fuzzy control systems as for a given fuzzy output a specific action ought to be taken in an automated engineering process. Nonetheless, when dealing with medical diagnosis, one wonders whether this is indeed required. Would not such procedure jeopardize the wealth of knowledge produced by fuzzy set theory and betray wisdom, or, in other words, would not defuzzification narrow the view one can get of a given subject instead of enlarging it?

In the present exercise, one could perhaps conclude that comparing the two models tested, both could be considered with a fair performance on predicting a patient's diagnosis since both have more correct than wrong hits. Max-min approach could be reckoned to some extent superior to Gödel's and the intermediate situation of being sick, but not bearing pneumonia could be deemed as poorly assessed by the set of clinical signs taken into analysis – indeed, under both models, classification of patients with other disease results rather challenging. Nonetheless, if defuzzification is left apart and results are offered as grades of membership to the different nosologic categories (table 2) much more information is provided to decision-making. Taking, for instance, the awkward case of patient n° 10, instead of allocating him to any given diagnosis by choosing any defuzzification rule, one would be much better off with the information that max-min suggests that he is equally weakly associated with the three diagnoses and that Gödel adds to this that he has some degree of a pathognomonic sign of pneumonia. Decision as whether his diagnosis is this or that should better be ultimated by medical judgement, which would use this as complementary information in its decision taking process.

Physicians endeavor to attain a diagnosis in order to trigger a set of corresponding actions concerned with re-establishing health, and these actions are the real final goal of medical care. Sadegh-Zadeh [4], giving emphasis to the fact that misdiagnosis might be as frequent as 40% in all patients seen, suggests that fuzzy medical systems should focus on actions to be taken instead of on diagnoses, which by their turn should be only one additional piece of information in the decision taking process. Considering this, it seems that a fuzzy medical system would better provide subsidiary information than experimenting with superseding the physician role.

Nguyen and Walker [5] , discussing defuzzification state that "*Standard control theory is efficient when precise mathematical models are available. The emergence of artificial intelligence technologies has suggested the additional use of manual control expertise in more complex problems.*" Medical systems are undoubtedly the case and last decision should be left to manual control expertise.

Medical sciences are concerned with causal relationships and, according with Susser [6], different frames of reference provide different views of such relations. Knowledge is completed through an exercise of bringing together different views and the present study was exemplary in this sense. With the information provided in table 1 one can learn, for instance, that chest X-ray and chest auscultation, in spite of being pathognomonic to pneumonia (Gödel's model), are not the clinical signs with the largest intersection with this diagnosis: dyspnoea, tachycardia and tachypnea (s_2, s_4, s_7) have much higher degrees of membership (max-min model). This is in agreement with the current medical belief that inspection of patients has precedence over radiological or laboratory investigation and reassures that, indeed, if either chest X-ray or auscultation is positive a diagnosis of pneumonia is very likely, but other clinical signs should not go unattended as they are as much or even more informative.

Likewise, finding out that the healthy status had some degree of membership to cardiac rate allowed the identification of a few cases that had been considered healthy, when they would have been better classified as bearers of some disease, either malnutrition or anemia. Further exploring this table, one can learn how each signal relate to each diagnosis and thus for which situation it is more important – for instance, toxaemia (s_6), even though not pathognomonic to pneumonia, is more important to this diagnosis than to the diagnosis of other disease.

Proceeding to table 2, where relations of each patient to each diagnosis are shown, one can realize that any patient is better described by his vector of relations with diagnoses than by a single allocation to a specific diagnosis. Table 4 provides evidence of this since it shows that if multiple allocation is allowed, the level of agreement is enhanced and no cases are left out of classification.

In conclusion, the results of this study were suggestive that fuzzy diagnostic systems should perhaps rule out defuzzification in favor of providing information about relations and leaving decision to medical judgement. Shirking defuzzification one should be avoiding the loss of wisdom that the poet warned against. Giving emphasis to relations one should be achieving real knowledge since according with Poincaré [7] "*outside relations there is no reality knowable*".

References

[1] KLIR, G. and YUAN, B.. Fuzzy Sets and Fuzzy Logic: Theory and applications. Prentice Hall, USA, 1995.

[2] PEDRYCZ W. and GOMIDE F.. An Introduction to Fuzzy Sets: Analysis and design. MIT Institute, USA, 1998.

[3] SANCHEZ E.. Medical diagnosis and composite fuzzy relations. Advances in fuzzy set theory and applications, 1979.

[4] SADEGH-ZADEH K.. Fundamentals of clinical methodology: 1. Differential indication. Artificial Intelligence in Medicine 1994; 6: 83-102.

[5] NGUYEN H.T., WALKER E.A.. A first course in fuzzy logic. 2nd Edition. New York: Chapman & Hall/CRC, 1999: 321.

[6] SUSSER M. Casual thinking in the health sciences. New York: Oxford University Press, 1973

[7] POINCARÉ, H. Science and hypothesis. New York: Dover Publications, 1952: xxiv.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Fuzzy rules in asymptomatic HIV virus infected individuals model

Rosana Sueli da Motta Jafelice
FAMAT - UFU / UNICAMP
rosanam@dca.fee.unicamp.br
Laécio Carvalho de Barros , Rodney Carlos Bassanezi
IMECC – UNICAMP
laeciocb@ime.unicamp.br, rodney@ime.unicamp.br
Fernando Gomide
FEEC – UNICAMP
gomide@dca.fee.unicamp.br

Abstract

The purpose of this paper is to study of the evolution of positive HIV population for manifestation of AIDS (Acquired Immunideficiency Syndrome). Our main interest is to model the transference rate in this stage. For this purpose, we use expert information on transference rate because it strongly depends of the viral load and of the level $CD4+$of the infected individuals. More specifically, the transference rate is modeled as a fuzzy set that depends of the knowledge on viral load and $CD4+$, respectively. The main difference between this model and the classic one relies on the biological meaning of the transference rate λ. Medical science uses linguistic notions to characterize disease stages and to specify anti-retroviral therapy. Fuzzy set theory provides the formal framework to model linguistic descriptions such as the transference rate λ using expert knowledge.

1. *INTRODUCTION*

In the last decade, the mathematical literature on imprecision and uncertainty has grown considerably, especially in system modeling, control theory, and engineering areas. More recently, several authors have used the fuzzy set theory in epidemiology problems [6] and population dynamics [2].

Since the advent of the HIV infection, several mathematical models have been developed to describe its dynamics [3] and [5]. In this paper fuzzy set theory, introduced in the sixties by Lotfi Zadeh [9], is used to deal with imprecision on the time instant or moment in which AIDS begins its manifestation. Our study considers the transference rate λ, as used in the classical Anderson's model [3], as a fuzzy set.

Fuzzy set theory is a genuine generalization of classical set theory [1], [7] and [9] useful to model unsharp classes. In this paper, the parameter λ is viewed as a linguistic variable whose values are fuzzy sets that depends on the viral load v and on the $CD4+$ level. $CD4+$ is the main T lymphocyte attacked by the HIV retrovirus when it reaches the bloodstream. The notion of λ rate as a linguistic variable with fuzzy values captures its biological meaning more faithfully [6] and [7] to classify the disease stages, and to decide on when the antiretroviral therapy should be used. We assume that, initially, the fraction of infected asymptomatic individuals x is maximum and equal to 1, and that the fraction of AIDS symptomatic individuals y is null. The next section introduces the main definitions needed in this paper.

2. *PRELIMINARY DEFINITIONS*

A *fuzzy subset* F of the universe set U is defined by a ***membership*** function u that assigns to each element x of U a number $u(x)$ between *zero* and one to mean the degree of membership from x to F. Therefore, the fuzzy subset (or *set,* for short) F is its membership function $u: U \to [0,1]$. In this paper we will use the membership function u to denote the fuzzy set F. It is interesting to observe that a classic subset A of U is a particular fuzzy set for which the membership function is the characteristic function of A, this is, $\chi_A : U \to \{0,1\}$. The main tool that we will rely on in this paper concerns fuzzy rule-based systems [7], whose architecture is shown in figure 1. The method of fuzzy inference chosen is referred to as the Mamdani method.

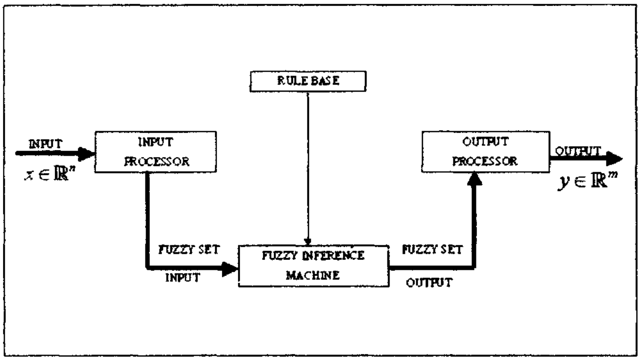


Figure 1: Architecture of fuzzy rule-based systems.

3. *CLASSIC MODEL*

The classical Anderson's model (1986) for AIDS is given by:

$$\frac{dx}{dt} = -\lambda(t)x \qquad x(0) = 1$$

$$\frac{dy}{dt} = \lambda(t)x = \lambda(t)(1-y) \qquad y(0) = 0 \qquad (1)$$

where $\lambda(t)$ is the transference rate between infected individuals and infected individuals that develop AIDS, x is the proportion of infected population that does not have AIDS symptoms yet, and y is the proportion of the population that has developed AIDS symptoms. Anderson assumes $\lambda(t) = at$, $a > 0$. Thus the solution of (1) is

$$x(t) = e^{-\frac{at^2}{2}} \qquad y(t) = 1 - e^{-\frac{at^2}{2}} \qquad (2)$$

4. *FUZZY MODEL*

When the HIV reaches the bloodstream, it attacks mainly the lymphocyte T of the $CD4+$ type. The amount of cells $CD4+$ in periferic blood has prognostics implication in infection evolution by HIV. Nowadays, the amount of immunecompetence cells is the most clinically useful and acceptable measure to treat of infected individuals by HIV, although it is not the only one. We can classify the amount of $CD4+$ cells/ml in the peripheral blood in four ranges (see: www.aids.gov.br):

1. **CD4+ > 0.5 cells/ml**: Stage of infection by HIV with low risk of to develop disease.
2. **CD4+ between 0.2 and 0.5 cells/ml**: Stage characterized for appearance of signs and shorter symptoms or constitutional alterations. Moderate risk to develop opportunist diseases.
3. **CD4+ between 0.05 e 0.2 cells/ml**: Stage with high possibility to develop opportunist diseases.
4. **CD4+ < 0.05 cells/ml**: High risk of get opportunist diseases such as Kaposi's sarcoma. High life risk and low survival chances.

On the other hand, a low HIV viral load not enough to destroy all the lymphocyte $CD4+$ of the organism. Thus, antibodies have chances to act against opportunist diseases. High viral load destroys large quantities of lymphocyte $CD4+$ and the immunologic system may lose its function.

In the beginning (or when change of anti-retroviral therapy occurs), the literature recommend viral load exams within one to two months period to evaluate the treatment. The results should be interpreted of the following way:

1. **Viral load below of 10.000 copies of RNA by ml**: low risk of disease progression.
2. **Viral load between 10.000 and 100.000 copies of RNA by ml**: moderate risk of disease progression.
3. **Viral load above of 100.000 copies of RNA by ml**: high risk of disease progression.

The identification of the disease stages and its respective treatment is based on the relation between viral load and $CD4+$ cells level. Control of the viral load and $CD4+$ cells level may interfere in the transference rate λ control.

Thus, the conversion from an asymptomatic individual to a symptomatic individual depends on the individual characteristics, as measured by the viral load v and $CD4+$. Therefore, we suggest the following modification of (1):

$$\begin{aligned} \frac{dx}{dt} &= -\lambda(v,CD4+)x & x(0) &= 1 \\ \frac{dy}{dt} &= \lambda(v,CD4+)x = \lambda(v,CD4+)(1-y) & y(0) &= 0 \end{aligned} \tag{3}$$

The difference between this and the first model (1) is that now the parameter $\lambda = \lambda(v, CD4+)$ has a clear biological meaning and thus is a more faithful characterization of λ. From the mathematical point of view we can think of (3) as a parametric family of systems. In this case, λ is the parameter dependent of v and $CD4+$. It seems reasonable that λ, and consequently of the population y, can be controlled via v and $CD4+$. From (3) we have

$$x(t) = e^{-\lambda(v,CD4+)t} \qquad y(t) = 1 - e^{-\lambda(v,CD4+)t}, \; t > 0 . \tag{4}$$

5. LINGUISTIC VARIABLES AND RULES BASE

Estimation of the transference rate $\lambda = \lambda(v, CD4+)$ is based on expert medical information. We adopt a fuzzy rule-based modeling approach assuming that the viral load (v), the $CD4+$ level, and the transference rate (λ) are linguistic variables [7]. According to section 4, the viral load is classified as low, medium and high whereas the $CD4+$ level is classified as very low, low, medium, high medium and high. The transference rate λ is classified as weak, medium weak, medium and strong. The $CD4+$ level between 0.2 and 0.5 cells/ml is divided in two ranges because it relates with an important phase of the transference of asymptomatic to symptomatic.

The fuzzy rule-based model uses the Mamdani inference method to derive the values of λ assuming the membership functions shown in figures 2, 3 and 4 and the following rule-base:

1. If v is low and $CD4+$ is very low then λ is strong.
2. If v is low and $CD4+$ is low then λ is medium.
3. If v is low and $CD4+$ is medium then λ is medium.
4. If v is low and $CD4+$ is high medium then λ is weak medium.
5. If v is low and $CD4+$ is high then λ is weak.
6. If v is medium and $CD4+$ is very low then λ is strong.
7. If v is medium and $CD4+$ is low then λ is strong.
8. If v is medium and $CD4+$ is medium then λ is medium.
9. If v is medium and $CD4+$ is high medium then λ is weak medium.
10. If v is medium and $CD4+$ is high then λ is weak.
11. If v is high and $CD4+$ is very low then λ is strong.
12. If v is high and $CD4+$ is low then λ is strong.
13. If v is high and $CD4+$ is medium then λ is medium.
14. If v is high and $CD4+$ is high medium then λ is medium.
15. If v is high and $CD4+$ is high then λ is medium.

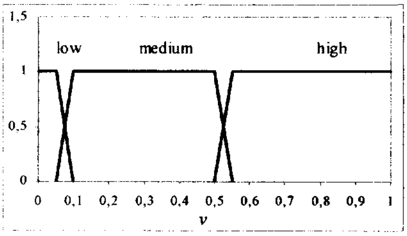


Figure 2: Membership functions viral load (v).

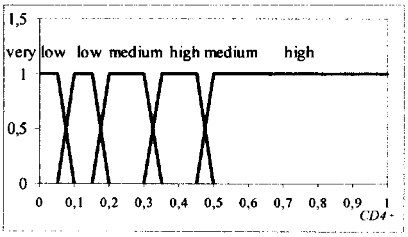


Figure 3: Membership functions CD4+ level.

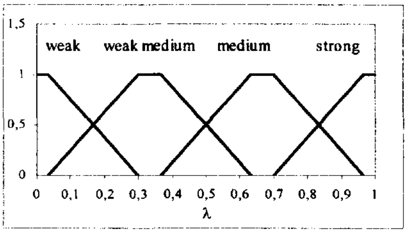


Figure 4: Membership functions lambda (λ).

6. *THE TRANSFERENCE RATE* λ

From of the rules base introduced in the previous section and from the inference method adopted, we simulate the viral load v and the $CD4+$ level to a HIV-positive individual during sixty months and to obtain the output $\lambda = \lambda(v, CD4+)$ as depicted in figure 5. A cross-section of the λ surface along a parallel plan to the $CD4+$ level axis is shown in figure 6.

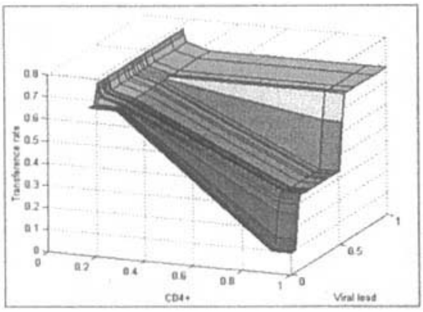


Figure 5: Function $\lambda = \lambda(v, CD4+)$.

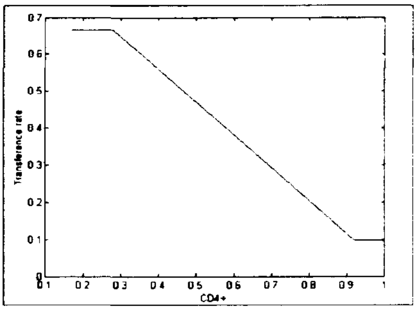
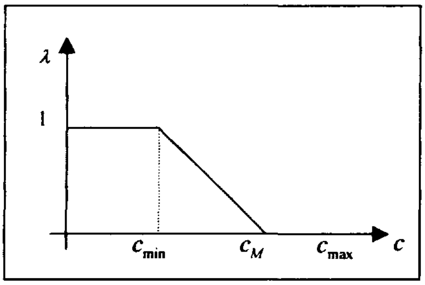


Figure 6: λ as a function of $CD4+$ ($v = 0.036$).

According with the section 4, $CD4+$ is the most useful parameter to control and to diagnose HIV. A more detailed study can be done assuming that the transference rate is $\lambda = \lambda(c)$, where $c = CD4+$ in the model (4). We assume λ given by the function of figure 6, that is, we assume $\lambda(c)$ as follows:

$$\lambda(c) = \begin{cases} 0 \quad \text{if} \quad c < c_{\min} \\ \dfrac{c_M - c}{c_M - c_{\min}} \quad \text{if} \quad c_{\min} \le c \le c_M \\ 1 \quad \text{if} \quad c_M < c < c_{\max} \end{cases} \tag{5}$$



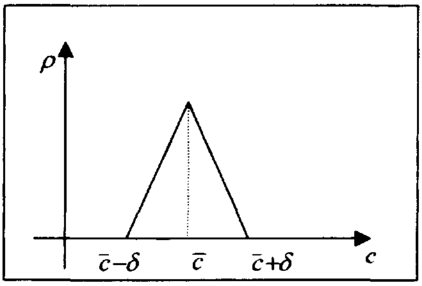


Figure 7: Transference rate λ in function of the $CD4+$. Figure 8: Membership function ρ adopted for C.

In the figure 7, $c_{\min}$ represents the minimum $CD4+$ level for which the individual becomes symptomatic, c_M represents the $CD4+$ level for which the chance to become symptomatic is minimum, and $c_{\max}$ is the largest quantity of $CD4+$ possible. In figure 6, we can observe the approximate values of c_{min} and c_M, this is, c_{min} is approximately 0.3 cells/ml and c_M is approximately 0.9 cells/ml. These values are compatible with the reality if $CD4+$ is less than 0.3 cells/ml the tendency is of the individual becomes symptomatic and when $CD4+$ is bigger than 0.9 cells/ml the tendency is of the individual to be asymptomatic. Thus, the number of asymptomatic and symptomatic individuals at the time instant t is:

$$x(t,c)=e^{-\lambda(v,CD4+)t} \qquad y(t,c)=1-e^{-\lambda(v,CD4+)t}$$

7. FUZZY EXPECTANCY OF THE ASYMPTOMATIC INDIVIDUALS

The fuzzy expectancy is introduced here as a deffuzification method to provide an average value of $x(t)=e^{-\lambda(v,CD4+)t}$, the number of asymptomatic individuals at the time instant t. To define fuzzy expectancy we need first to define a fuzzy measure. Let Ω an unempty set and $P(\Omega)$ the subsets of Ω. The function $\mu: P(\Omega)\rightarrow[0,1]$ is a fuzzy measure [4] if:

1. $\mu(\varnothing)=0$ and $\mu(\Omega)=1$.
2. $\mu(A)\leq\mu(B)$ *if* $A\subseteq B$.

The value of the fuzzy expectancy of the asymptomatic individuals of the fuzzy set $x=x(t,c)$ is given by [8]

$$FEV[x]=\sup_{0\leq\alpha\leq 1}\inf[\alpha,\mu\{x\geq\alpha\}]$$

where $\{x\geq\alpha\}=\{c:x(c)\geq\alpha\}$ for each t and μ is a fuzzy measure. Let $H(\alpha)=\mu\{c:x(c)\geq\alpha\}$, for each $t>0$. Here we suggest the following fuzzy measure:

$$\mu(A)=\begin{cases}\sup_{c\in A}\rho(c) & if \quad A\neq\varnothing\\ 0 \quad if \quad A=\varnothing\end{cases}.$$

Let $A=[a,c_{\max}]$, where $a=c_M-(c_M-c_{\min})\left(\dfrac{-\ln\alpha}{t}\right)$. Observe that $c_{\min}<a\leq c_M$. Note that μ is an optimist measure, in the way that the $CD4+$ level in a group is evaluated for individual with $CD4+$ best level.

Beyond of $c\in[0,c_{\max}]$, we assume that the $CD4+$ level of the group HIV-positive studied (C) has different possibility to occur. We assume c as a triangular fuzzy set (see figure 8) given by:

$$\rho(c)=\begin{cases}0 \quad if \quad c\leq\bar{c}-\delta\\ \dfrac{1}{\delta}(c-\bar{c}+\delta) \quad if \quad \bar{c}-\delta<c\leq\bar{c}\\ \dfrac{-1}{\delta}(c-\bar{c}-\delta) \quad if \quad \bar{c}<c\leq\bar{c}+\delta\\ 0 \quad if \quad c>\bar{c}+\delta\end{cases} \qquad (6)$$

The parameter $\bar{c}$ is the modal value and δ is the dispersion of each one the set fuzzy that defines the linguistic variable values. These fuzzy sets are defined from of the values $c_{\min}$, c_M and $c_{\max}$ of the definition of λ. Thus, we the fuzzy expectancy by looking at three cases:

1. **Case:** $CD4+$ low (C_-). We assume $c_{\min}>\bar{c}+\delta$. Thus, $FEV[x]=e^{-t}$.

2. **Case:** $CD4+$ high (C^+). We assume $c_M \leq \overline{c} - \delta$ and $\overline{c} + \delta \leq c_{\max}$. Thus, $FEV[x] = 1$.

3. **Case:** $CD4+$ medium (C_-^+). We assume $\overline{c} - \delta > c_{\min}$ and $\overline{c} + \delta < c_M$. The third case is the most interesting for our purposes because most part of the individuals is within this range. After a number of manipulations, we obtain:

$$H(\alpha) = \begin{cases} 1 & if \quad 0 \leq \alpha \leq e^{-\lambda(\overline{c})t} \\ \rho(a) & if \quad e^{-\lambda(\overline{c})t} < \alpha < e^{-\lambda(\overline{c}+\delta)t} \\ 0 & if \quad e^{-\lambda(\overline{c}+\delta)t} \leq \alpha \leq 1 \end{cases} \tag{7}$$

where $\rho(a) = \frac{1}{\delta}\left[-c_M - (c_M - c_{\min})\left(\frac{\ln \alpha}{t} \right) + \overline{c} + \delta \right]$. As $H(\alpha)$ is continnuos and decreasing, it has a unique fixed point that coincides with $FEV[x]$. Thus we obtain the following inequality:

$$e^{-\lambda(\overline{c})t} < FEV[x] < e^{-\lambda(\overline{c}+\delta)t}. \tag{8}$$

The inequality above shows that in the best hypothesis (μ optimist) is possible the $FEV[x]$ is less than 1, since it is inferior a possible solution $e^{-\lambda(\overline{c}+\delta)t}$.

This way, for each $t > 0$ there exists a unique $c(t) \in (\overline{c}, \overline{c} + \delta)$, where $c(t) = c_M + (c_M - c_{\min})\left(\frac{\ln \alpha(t)}{t} \right)$. Thus, $FEV[x] = e^{-\lambda(c(t))t}$ is an exponential curve and, because the $CD4+$ level increases with t, $FEV[x]$ is decreasing. Consequently, the fuzzy expectancy is not solution of (3). Actually, at each instant t, the $FEV[x]$ coincides with the unique solution of (3). It is easy to verify that $FEV[x]$ is differentiable and that it satisfies the following differential equation with the time dependent parameter λ:

$$\frac{dx}{dt} = -\left[\lambda(c(t)) + t\frac{d\lambda}{dt}(c(t))\frac{dc}{dt}(t) \right]x \tag{9}$$

Note that for the three cases studied here, we obtain the following inequality:

$$e^{-FEV[\lambda]t} \leq e^{-\lambda(\overline{c})t} \leq FEV[x]. \tag{10}$$

8. CONCLUSION

The main difference between the deterministic and the fuzzy model is that the fuzzy model allows imprecise parameters and deffuzification at any time. In deterministic modeling, deffuzification is done in the beginning of the mathematical modeling. We may state that the deterministic models are particular instances of fuzzy models. In our case, the deterministic model (3) results the solution $x(t) = e^{-\lambda(\overline{c})t}$. The fuzzy model also provides a unique curve $FEV[x]$ when decisions are necessary. Fuzzy expectancy of *asymptomatic* individuals is bounded below by $e^{-\lambda(\overline{c})t}$. Therefore, the fuzzy model over evaluates the number of *asymptomatic* individuals. We emphasize that, despite of using an optimist fuzzy measure to evaluate the $CD4+$ level in the population, the $FEV[x]$ is less than $e^{-\lambda(\overline{c}+\delta)t}$ and thus depends of the populational dispersion δ of the $CD4+$ level of the group studied. From (8), we see that when $\delta \to 0$, $FEV[x] \to e^{-\lambda(\overline{c})t}$ that is, it depends only of $CD4+$, indicating a policy for treatment. Clearly, the fuzzy model provides a clearer and meaningful characterization of parameter λ that is compatible with medical knowledge and perception.

9. REFERENCES

[1] Barros, L.C., R.C. Bassanezi and M.B. Leite. The Epidemiological Models SI with Fuzzy Parameter of Transmission (submitted).
[2] Krivan, V. and G. Colombo. A Non-Stochastic approach for Modeling Uncertainty in Population Dynamics. Bulletin of Mathematical Biology (1998) 60 .
[3] Murray, J.D. 1990. Mathematical Biology. Springer-Verlag, Berlin.
[4] Nguyen, H.T. and E.A.Walker. 2000. A First Course in Fuzzy Logic. Chapman & Hall/CRC.
[5] Nowak, M.A . 1999. The Mathematical Biology of Human infections. Conservation Ecology 3 .
[6] Ortega, N.S. 2001. Aplicação da Teoria de Lógica Fuzzy a Problemas da Biomedicina., PhD thesis., University of São Paulo (in Portuguese).
[7] Pedrycz, W. and F.Gomide. 1998. An Introduction to Fuzzy Sets: Analysis and Design. Massachusetts Institute of Technology.
[8] Sugeno, M. 1974. Theory of Fuzzy Integrals and Its Applications, PhD thesis, Tokyo Institute of Technology, Japan.
[9] Zadeh, L.A . 1965. Fuzzy Sets. Information and Control 8: 338-353.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Categorical Limits and Reuse of Algebraic Specifications

Isabel Cafezeiro* and Edward Hermann Haeusler**
**Department of Informatics*
Universidade Federal Fluminense
Niterói Brazil
e-mail: isabel@dcc.ic.uff.br
Bolsista CNPq - Brasil
***Department of Informatics*
Pontifícia Universidade Católica
Rio de Janeiro, Brazil
e-mail: hermann@inf.puc-rio.br

Abstract. In this paper, we present a definition of limits in the category of signatures. We extend this definition to the category of specifications showing how limits can be used as another modular way of combining specifications. An example shows how reuse can be achieved from limits.

1 Introduction

It is very common the use of Category Theory and its constructors in the study of Algebraic Specifications. In what concerns the construction of modular specifications, the colimit operation plays a special role [3]: it makes possible to reduce a gamma of traditional ways of composing specifications into a single operation. Parameterization, renaming of components and sum of specifications are examples of operations that can be well described by colimits. Limits however do not appear very frequently as combinator of specifications. Their role uses to be restricted to combinators of sorts. For example, in [7] we can find the product of sorts as one of the basic operations. A possible justification for this fact is that, when concerning the modular construction of specifications, we first think about using pre-existing specifications to build richer ones. For this task the colimit is the appropriated operation. But the opposite direction also seems to be very fruitful: to build less informative specifications from pre-existing richer ones. Consider, for example the case where we have two specifications: equivalence relation and partial order. We want to specify another relation that has a common intersection with these pre-existent ones: it is also a pre-order. Thus we just need a way of extracting what these two specifications have in common. We claim that limit is the appropriated operation to perform this task.

In this paper, we present a definition of limits in the category of signatures. We extend this definition to the category of specifications showing how limits can be used as another modular way of combining specifications. Finally, we show by example how inheritance and reuse can be achieved from limits.

The paper is organised as follows: in section 2 we present the basic definitions of signatures and the category *Sign* of signatures. In 3 we define limits in *Sign*. Section 4 presents definitions of specifications and the corresponding category *Spec*. In 5 we define limits in *Spec* and presents an algorithm to ensure that the text of specifications will be finite. In section 6 we present the example to illustrate reuse. Finally, in 7 we conclude and comment related works.

2 Algebraic Signatures

We adopt here a notation that slightly differs form that of [5] for algebraic signatures. We put together ranks of the same size. Signature morphisms, however, are restricted to ranks (as before) and not to ranks size.

Definition 1 *An algebraic signature* Σ *is a pair* $< S, F >$ *where* S *is a set (of sort names) and* F *is an indexed family of sets (of operator names). For a given* $f : u \rightarrow s \in F_n$, *where* $u \in S^*$ *and* $s \in S$ $|us|$ *(the length of us) is equal to n. We say that* $f : u \rightarrow s$ *has rank us, and we denote* F_{us} *the subset of* F_n *whose members have rank us.*

Definition 2 *Given* $\Sigma =< S, F >$ *and* $\Sigma' =< S', F' >$, *algebraic signatures, a signature morphism* $\sigma : \Sigma \rightarrow \Sigma'$ *is a pair* $< \sigma_S : S \rightarrow S', \sigma_F >$ *where* σ_S *is a function and* $\sigma_F =< \sigma_{F^n} : F^n \rightarrow F'^n >$ *is a family of functions, such that,*
- *if* $\sigma_{F^n}(f) = f'$ *then* $rank(f) = us$ *and* $rank(f') = \sigma_S^*(u)\sigma_S(s)$;
- *if* $F^n \neq \emptyset$ *then* $\sigma_F(F^n) \neq \emptyset$.

The function *rank* returns the rank of a function, or of a set of functions.

$\sigma_{F_{us}}$ refers to the restriction of σ_{F_n} whose domain is the set of functions with rank us.

The collection of algebraic signatures and signature morphisms form a category which will be called *Sign*.

3 Limits in the Category of Signatures

In this section we define limits in *Sign* and exemplify with the particular case of products. In [1] the reader can find descriptions of particular kinds of limit in the category of signatures, together with illustrative examples.

Recall from the definition of morphisms in Σ that, in order to have a morphism between two signatures it is necessary to have a function σ_{F_k} for each rank length k from the domain signature. Moreover, for each us from the domain signature, there must have at least one operation name in the rank $\sigma_S^*(u)\sigma_S(s)$. For example, there is not a morphism between the two signatures $\Sigma_1 =< \{a, b, c\}, < \{F_{abc}, F_{aaa}\} >>$ and $\Sigma_2 =< \{p, r\}, < \{F_{ppr}\}, \{F_{pr}\} >>$.

From Σ_1 to Σ_2, an attempt to define $\sigma_{\{a,b,c\}}$ could be linking a and b to p and c to r. In this way, it would be possible to define $\sigma_{F_{abc}} : F_{abc} \rightarrow F_{ppq}$. But $\sigma_{F_{aaa}} : F_{aaa} \rightarrow F_{ppp}$ would remain undefined as F_{ppp} does not exists.

In the opposite side, the option would be linking both p and r to a so that we could define $\sigma_{F_{ppr}} : F_{ppr} \rightarrow F_{aaa}$. But, again, the rank pr would not have a corresponding aa in Σ_1.

Now, we can define the limit of a diagram in *Sign* component by component: the limit of sorts and the limit of each rank length that appears in all the signatures that compose the diagram. The operation names whose rank length is in some, but not in all the signatures will not be present in the limit signature.

Definition 3 *Let* $\Sigma = < S, F >$ *and* $\Sigma_i = < S_i, F_i >$. *Given a diagram* D *with objects* Σ_i *and morphisms* φ_j, *a* D*-cone* $\rho_i : \Sigma \to \Sigma_i$ *is a limit of D if and only if,*
$\rho_{i_S} : S \to S_i$ *is the limit of sorts, and,*
F *is the family of limits* $\rho_{i_{F_k}} : F_k \to F_{i_k}$ *for each* k, *rank length of all the signatures* Σ_i.

To see that the above definition is indeed the limit in *Sign* we must verify the universal property of limits. Consider another D-cone $\rho'_i : \Sigma' \to \Sigma_i$ that could also be a limit of D. By the definition of morphisms in *Sign* and by the fact that both Σ and Σ' are vertexes of D, they do not have a rank length k that is not present in all the signatures Σ_i. Thus Σ and Σ' have at most the rank lengths that are present in all the signatures of D. Then, either Σ and Σ' have the same rank lengths or Σ has some rank lengths that Σ' does not have. In both cases, the unique morphism from Σ' to Σ is ensured by the universal property component by component (sorts and each rank length).

We state the universal property of limits:

> *Given a diagram* D *with objects* Σ_i *and morphisms* φ_j, *and the cone* $\rho : \Sigma \to \Sigma_i$, *a limit for* D. *For any other cone* $\rho' : \Sigma \to \Sigma_i$, *there exists a unique morphism* $\tau : \Sigma' \to \Sigma$ *such that* $\rho_i \circ \tau = \rho'_i$, *for each* i, *index of object of* D.

The universal property restated for a diagram without morphisms and just two signatures characterises the universal property of a product:

> *Given a diagram* D *with* Σ_1 *and* Σ_2 *as objects and without morphisms. Let* Σ, *be a limit for* D *(thus, there are morphisms* $\pi_1 : \Sigma \to \Sigma_1$ *and* $\pi_2 : \Sigma \to \Sigma_2$*). For any other* Σ' *with morphisms* $\rho'_1 : \Sigma' \to \Sigma_1$ *and* $\rho'_2 : \Sigma' \to \Sigma_2$, *there exists a unique morphism* $\tau : \Sigma'$ *to* Σ *such that* $\pi_1 \circ \tau = \rho'_1$ *and* $\pi_2 \circ \tau = \rho'_2$.

Note that, to play the role of Σ' a signature must have at most the rank lengths that are present in both Σ_1 and Σ_2. Otherwise, the morphisms $\rho'_1 : \Sigma' \to \Sigma_1$ and(or) $\rho'_2 : \Sigma' \to \Sigma_2$ would not exist.

As an example, the product of the two signatures $\Sigma_1 = < \{a, b, c\}, < \{F_{abc}, F_{aaa}\} >>$ and $\Sigma_2 = < \{p, r\}, < \{F_{ppr}\}, \{F_{pr}\} >>$ would be $\Sigma_1 \times \Sigma_2 = < \{ap, aq, bp, bq, cp, cq\}, < \{F_{abc} \times F_{ppr}, F_{aaa} \times F_{ppr}\} >>$.

4 Algebraic Specifications

We transcribe the usual definitions of algebraic specifications and specifications morphisms.

Definition 4 *An algebraic specification is a pair* $Sp = < \Sigma, \Phi >$ *where* Σ *is an algebraic signature and* Φ *is a set of equations constructed with the symbols of* Σ, *called* Σ*-equations.*

Definition 5 *Given* $Sp = < \Sigma, \Phi >$ *and* $Sp' = < \Sigma', \Phi' >$, *algebraic specification, an specification morphism from* Sp *to* Sp' *is a pair* $< \sigma : \Sigma \to \Sigma', \alpha >$ *such that* σ *is a signature morphism and* $\alpha : \Phi \to \Phi'$ *exists if and only if for all* $e \in \Phi, \Phi' \vdash Sen(\sigma)(e)$.

In the above definition, *Sen* is a functor from the category *Sign* to *Set* that maps a signature to the set of all sentences that can be written in that signature. Sentences (equations) are pairs of terms of the same sort denoted by $t_i = t_j, i, j < \omega$. Terms of a signature are those obtained by the application of the two following items:

For a given signature $< S, F >$, and a set X_s of variables for each $s \in S$

(i) A variable $x \in X_s$ is a term of sort s;

(ii) If $t_1, ..., t_n$ are term of sorts $s_1, ..., s_n$, and $f : s_1, ..., s_n \rightarrow s$ is an operation symbol of the signature then $f(t_1, ..., t_n)$ is a term of sort s.

When applied to a morphism $\sigma : \Sigma \rightarrow \Sigma'$, the functor Sen gives a function $Sen(\sigma) : Sen(\Sigma) \rightarrow Sen(\Sigma')$ that translates each sentence of the signature Σ to a sentence of the signature Σ'. By $Sen(\sigma)(e)$ we mean the translation of an equation e of Φ to the signature Σ'. By $Sen(\sigma)(\Phi)$ we mean the translation of all equations of Φ to the signature Σ'. Thus, given two specifications, there is a morphism linking the first to the second if and only if the translation of each equation of the first can be proved in the second.

The collection of algebraic specifications and specifications morphisms form a category which will be called *Spec*.

5 Limits in the Category of Specifications

In this section we show that *Spec* has limits in general. In [2] the reader can find descriptions of particular kinds of limit in the category of specifications together with illustrative examples.

Limit in *Spec* will be constructed as pairs of limits: in the signature (as defined in 3) and in the set of equations.

Let us call $Cn(\Phi)$ the set of provable equations from Φ, by the following set of rules:

$(r) : x_s = x_s$

$(s) : t_1 = t_2 \longrightarrow t_2 = t_1$

$(t) : t_1 = t_2, t_2 = t_3 \longrightarrow t_1 = t_3$

$(cong) : t_1 = t'_1, ..., t_n = t'_n \longrightarrow f(t_1, ..., t_n) = f(t'_1, ..., t'_n)$

$(subs) : t_1(y) = t_2(y), t_3 = t_4 \longrightarrow t_1(t_3) = t_2(t_4)$

In (r), for each sort s, x_s is a distinguished variable of s used only to state the reflection rule. The usual reflection rule is obtained from this by the application of $(subs)$. In $(cong)$, f is any operation symbol and the sequence of terms is consistent with the rank of f. In $(subs)$, y is a variable of the same sort of t_3 and t_4. The notation $t(y)$ in the left means that y appears at least once in t, and the notation $t_i(t_j)$ means that one instance of y is replaced by t_j in t_i.

Definition 6 *Let* $Sp =< \Sigma, \Phi >$ *and* $Sp_i =< \Sigma_i, \Phi_i >$. *Given a diagram D with objects* Sp_i *and morphisms* $< \varphi_j, \alpha_j >$, *a D-cone* $< \rho_i, \beta_i : \Phi \rightarrow \Phi_i >: Sp \rightarrow Sp_i$ *is a limit of D if and only if,*

ρ_i *is the limit of signatures, and,*

$\Phi = \bigcap \Psi_i$, *where* $\Psi_i = \{e | Sen(\sigma_i)(e) \in Cn(\Phi_i)\}$.

The set Φ is intended to be just the intersection of the consequences of Φ_i, but as the sets Φ_i may not be written in the same signature, we will need a translation provided by $Sen(\sigma_i)$.

Recall that $\sigma_i : \Sigma \rightarrow \Sigma_i$ may not be injective, as usual in projections. Thus, for a symbol in Σ_i we may have many corresponding symbols in Σ. As a consequence the number of axioms in the limit specification will grow a lot. In addition, σ_i may not be surjective. In this case, some symbols in Σ_1 may not have any corresponding in Σ. This is not a problem, as axioms written with them would not be proved in the other specifications, thus they do not belong to the limit.

We assume that $Cn(\Phi_i)$ is decidable, otherwise they would not be of practical computational use. The set of equations of the specifications may not be finite, but must be recursive. The intersection Φ is also decidable, and thus, can be recursively described. This is enough for considering Sp an specification. It would be desirable, however, that we could construct this object. It is a very common situation in the practice of algebraic specifications that objects have both Σ_i and Φ_i finite. In the following we describe the construction of the limit in such case.

Consider a diagram with objects $SP_i =< \Sigma_i, \Phi_i >$ and morphisms $< \phi_i, \alpha_i >$, where Σ_i and Φ_i are finite.

Let us enumerate the set of sentences in Φ in the same way we enumerate strings: by size, and for string of same size, by a lexicographic order based on the enumeration of symbols of $\Sigma \cup V$, where V is a sufficiently big finite set of variables for each sort. Consider the following enumeration of subsets of Φ:

$$\Gamma_0 = \emptyset$$
$$\Gamma_n = \begin{cases} \{e_n\} \cup \Gamma_{n-1}, & \text{if } \Gamma_{n-1} \nvdash e_n; \\ \Gamma_{n-1}, & \text{otherwise.} \end{cases}$$

To get finite set of equations that axiomatize Φ we follow this chain until reaching a set Γ which has the following property:

$(*)$ Γ contains all equations e of $\bigcup c$ that are present in $\bigcap\{e|Sen(\sigma_i)(e) \in \Phi_i\}$.

Note that Γ is not just, the set of equations that belongs to the union of specifications (axioms) and can be proved from all specifications, considering the appropriated translations. It also includes equations that can be proved in all specifications but are not axioms.

In addition, Γ solves the problem of equivalent, but syntactically different equations: if e is an axiom of SP_1, e' is na axiom of SP_2 and $e \neq e'$ but e is equivalent to e', then, for $i = 1, 2$, $Cn(\bigcap\{e|Sen(\sigma_i)(e) \in \Phi_i\})$ would not give the correct set of consequences. $\bigcap\{e|Sen(\sigma_i)(e) \in Cn(\Phi_i)\}$ gives the desired set, but is infinite. Syntactical differences are solved by the enumeration of Γ_i because the larger of equivalent equations will be present in Γ (by $(*)$), and thus, all equations equivalent equation of least size will also be in Γ.

Proposition 1 shows that Γ is indeed the limit.

Proposition 1 *$\Gamma \vdash e$ if and only if $e \in \bigcap\{e|Sen(\sigma_i)(e) \in Cn(\Phi_i)\}$*

$(\rightarrow)$ Members of Γ come from $\bigcap\{e|Sen(\sigma_i)(e) \in Cn(\Phi_i)\}$.

$(\leftarrow)$ Induction on the size of proofs of equations in $\bigcap\{e|Sen(\sigma_i)(e) \in Cn(\Phi_i)\}$. Let us denote a proof of an equation e $\bigcap\{e|Sen(\sigma_i)(e) \in Cn(\Phi_i)\}$ by P_e, and its size by $|P_e|$. Select a proof P_e such that $|P_e| = 1$. Then e belongs to one of the sets Ψ_i, hence, by $(*)$, $e \in \Gamma$. Suppose that, for a proof P_e such that $|P_e| \leq k$ it is a fact that $\Gamma \vdash e$. Now, we show that for any $P_{e'}$ such that $|P_{e'}| = k + 1$, we have $\Gamma \vdash e'$. As $|P_{e'}| = k + 1$ then the first k steps form a proof of an $e \in \bigcap\{e|Sen(\sigma_i)(e) \in Cn(\Phi_i)\}$, and the last step is obtained from the previous ones by application of (r), (s), (t), $(cong)$ or $(subs)$. We analyse each of these cases:
(r) e' is $t = t$, then $\Gamma \vdash e'$;
(s) As $|P_{e'}| = k + 1$, there exists e with $|P_e| \leq k$ such that $P_{e'}$ is obtained from P_e by (s). By hypothesis, since $|P_e| \leq k$, $\Gamma \vdash e$. Then, by (s), $\Gamma \vdash e'$;
(t) As $|P_{e'}| = k + 1$, there exists e_1, e_2 with $|P_{e_1}| \leq k$ and $|P_{e_2}| \leq k$ such that $P_{e'}$ is obtained

from P_{e_1} and P_{e_2} by (t). By hypothesis, since $|P_{e_1}| \leq k$ and $|P_{e_2}| \leq k, \Gamma \vdash e_1$ and $\Gamma \vdash e_2$. Then, by (t), $\Gamma \vdash e'$;
$(cong)$ As $|P_{e'}| = k + 1$, there exists e with $|P_e| \leq k$ such that $P_{e'}$ is obtained from P_e by $(cong)$. By hypothesis, since $|P_e| \leq k, \Gamma \vdash e$. Then, by $(cong)$, $\Gamma \vdash e'$;
$(subs)$ As $|P_{e'}| = k + 1$, there exists e_1, e_2 with $|P_{e_1}| \leq k$ and $|P_{e_2}| \leq k$ such that $P_{e'}$ is obtained from P_{e_1} and P_{e_2} by $(subs)$. By hypothesis, since $|P_{e_1}| \leq k$ and $|P_{e_2}| \leq k, \Gamma \vdash e_1$ and $\Gamma \vdash e_2$. Then, by $(subs)$, $\Gamma \vdash e'$.

6 Limits and Reuse of Specifications

We turn now to the example cited in the introduction of this text to show how limit is a promising operation to achieve reuse of specifications.

We transcribe below the specifications of equivalence relation and partial order. The boolean part of this specifications could be specified in separate, as shows [3], but we are avoiding the use of operations of modularization other than limit.

	Equivalence relation (Sp_{ER})		Partial order (Sp_{PO})
Sorts:	B, S	Sorts:	B, S
Opns:	$f : S \times S \to B$	Opns:	$f : S \times S \to B$
	$T : B$		$T : B$
	$F : B$		$F : B$
	$\wedge : B \times B \to B$		$\wedge : B \times B \to B$
	$\neg : B \to B$		$\neg : B \to B$
	$\to: B \times B \to B$		$\to: B \times B \to B$
			$\sim: S \times S \to B$
Eqns:	$f(a,a) = T$	Eqns:	$f(a,a) = T$
	$f(a,b) = f(b,a)$		$(f(a,b) \wedge f(b,a) \to a \sim b) = T$
	$(f(a,b) \wedge f(b,c) \to f(a,c)) = T$		$(f(a,b) \wedge f(b,c) \to f(a,c)) = T$
	$\neg T = F$		$\neg T = F$
	$\neg\neg b = b$		$\neg\neg b = b$
	$b \wedge T = b$		$b \wedge T = b$
	$b \wedge F = F$		$b \wedge F = F$
	$a \to b = a \wedge \neg b$		$a \to b = a \wedge \neg b$

The limit specification will have as sorts the product of sorts of both signatures, thus, $\{BB, BS, SB, SS\}$. The operations will be also the product of operations, but respecting the length of ranks. We have, thus, four operations of rank length one: $\{TT : BB, TF : BB, FT : BB, FF : BB\}$. Of rank length two, we have just one operation $\{\neg\neg : BB \to BB\}$. Finally, of rank length three, we have $\{ff, f\wedge, f \to, f \sim, \wedge f, \wedge\wedge, \wedge \to, \wedge \sim, \to f, \to \wedge, \to\to, \to\sim, \}$. The sorts and equations whose names are composed by different names are irrelevant for the application and can be easily eliminated by restriction operations. What is relevant for the application is to show that ff, the operation that corresponds to f in both specifications is still reflexive and transitive but is not symmetric or anti-symmetric.

For each Φ_i, the set Ψ_i is the set of equations written in Σ that, when translated by σ_i belong to the theory of specification i. When translated by σ_j, for $j \neq i$, it gives a set of equations that belong to the theory of specification j. Thus, $\bigcap \Psi_i$ is the intersection of theories, expressed in the language Σ. Then considering that $f(a,a) = T$ is in both Sp_{ER} and Sp_{PO}, there will be in the limit Sp the axiom $ff(aa,aa) = TT$. In the same way, the transitive law will be present in Sp. The anti-symmetric law can not be expressed in the limit Sp, as the symbol $\sim$ has not a corresponding in Sp_{ER}. The equation $f(a,b) = f(b,a)$ of Sp_{EQ} can not have a corresponding $ff(aa,bb) = ff(bb,aa)$ in the limit Sp as the translation by σ_{PO} results in $f(a,b) = f(b,a)$ that can not be proved in Sp_{PO}.

7 Conclusion and Related Works

To achieve reuse of specifications, we have to be able of building abstract specifications from concrete ones. The abstract specification may, then, give raise to other concrete specifications that share properties with the original ones. We showed in this paper that the limit will generate a specification that keeps the desired common properties and excludes the not common ones. Unfortunately, it is just the fact of having the same rank length that determines the presence of operation names of different specifications as a single operation name in the limit. As an example, $f : S \times S \rightarrow B$ of Sp_{EQ} and $\wedge : B \times B \rightarrow B$ of Sp_{PO} will generate in Sp the operation, $f\wedge : SB \times SB \rightarrow BB$ with the appropriated axioms in Sp. As $f\wedge$ represents the union of different concepts it may be excluded from the limit by a restriction operation. It is seems to be common that applications of limit require additional restriction operations. For example, limit is used in [8] to perform the parallel composition of transition systems. The limit operation includes unwanted transitions, so it is necessary the use a synchronisation table where the transitions to be eliminated are specified. The result of limit is restricted by this table. Also in [4] we can find similar uses of the limit, but in the context of sequential automata. In addition, the author suggests a categorical version of the synchronisation table based in colimit operation.

As we stressed in the introduction, it is not very easy to find studies of limits as an operation to combine specifications. In [6] we can see a complete presentation of the definitions of limits for specifications based on indexed categories. The author does not comment, however, any application of the concepts. Another point is that when defining the limit of theories, he is not concerned in keep finite the size of the specifications that are obtained from finite specifications. The approach presented here could also have been described as indexed categories, but in a different way of [6], as the indexes would be the rank lengths.

References

[1] I Cafezeiro and E. H. Haeusler. Limits and other Categorical Concepts in Algebraic Signatures. Toward Limits of Algebraic Specifications. Technical report, Uff, Rio de Janeiro, 2002.

[2] I Cafezeiro and E. H. Haeusler. Limits in the Category of Algebraic Specifications. Technical report, Uff, Rio de Janeiro, 2002.

[3] H. Ehrig. Introduction to Algebraic Theory of Graph Grammars. In V. Claus and G. Rozenberg, editors, 1st *Graph Grammars Workshop*, pages 1 – 69. Springer Verlag, 1979. Lecture Notes in Computer Science. 73.

[4] P. F. Blauth Menezes. *Reificação de Objetos Concorrentes*. PhD thesis, Instituto Superior Tcnico - Tese de Doutorado, Lisboa, Portugal, 1997.

[5] A. Tarlecki. Algebraic preliminaries. In E. Artesiano, H. J. Kreowski, and B. Krieg-Bruckner, editors, *Algebraic Foundations of Systems Specification*. Springer, Berlin. 1999.

[6] A. Tarlecki, J. A. Goguen, and R.M. Burstall. Some fundamental algebraic tools for the semantics of computation. part iii: Indexed categories. *Theoretical Computer Science 91*, 1991.

[7] R. Waldinger, Y.V. Srinivas, Goldberg A., and R. Jullig. *Specware Language Manual*. Suresoft,Inc, Kestrel, 1996.

[8] G. Winskel and M. Nielson. Categories in Concurrency. In Andrew M. Pitts and Peter Dybjer, editors, *Semantics and Logics of Computation*. Cambridge University Press, Cambridge, 1997.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Constructive Program Synthesis using Intuitionist Logic and Natural Deduction

SILVA,G.M.H* - HAEUSLER, E.H* - VELOSO, P.A.S+
**Dept. of Informatics, PUC-Rio, Brazil*
+Systems and Comp. Engin. Prog., COPPE and Comp. Sci. Dept., Inst. Math., UFRJ-Brazil
Email: hamazaki,hermann@inf.puc-rio.br

Abstract. We present a method to extract programs from constructive derivations, which is known as constructive synthesis or proof-as-program [2]. This method comes from the *Curry-Howard* isomorphism [11] and is based on the fact that a constructive proof for a theorem, which describes a problem, can be seen as a description of the solution of a problem, i.e., an algorithm [10,15]. In contrast with other constructive program synthesizers, in our work, the program (in an imperative language) is generated from a proof in many-sorted intuitionist logic using, as deductive system, the Natural Deduction. In addition, we provided a proof that the program generated is a representation of the solution for the specified problem by the theorem, in any theory that describes the data types used.

1 Introduction

Software development has to face two major problems: the cost of non-standard software - caused by the development times and the constant need for maintenance - and the lack of confidence in the reliability of software [13]. Many researchers are interested in providing techniques for developing reliable software, which is guaranteed to be correct and documented in a way that is easy to maintain and adapt. One of these research areas is called program synthesis, which proposes to generate automatically a correct program from specifications ([4,3,9,6,1]).

There are three basic categories of program synthesis: proof-as-program, transformational synthesis[1][14,7] and knowledge based program synthesis [13]. Some authors [14,7] insert another category called inductive program synthesis

Here, we deal with the proof-as-program paradigm [2], which avoids the double work of the software designing - the implementation of the system and the program verification - which can be seen as the same programming process in different degrees of formality. So this paradigm has focused on developing a program and its correctness proof at the same time [9,3].

This idea is based on the fact that: 1- Developing a program and prove that it is correct are just two aspects of the same problem [8]; 2- A proof for an application may be regarded as a (constructive) solution for a problem [17]; 3- A program can be extracted from a (constructive) proof of the existence of a solution for the corresponding problem [3].

Thus, using formal proofs - as a method for reasoning about the specification - and proving that the extraction process of a program preserves the proof's semantics, we get an automated way to construct, from a mathematical specification, a program that is correct by construction.

---

[1] Also called deductive program synthesis

The specification of the problem and the deductive rules provide information about the algorithm structures and the reasoning about their properties. There are many formal logical calculi to represent it properly, e.g., ITT (*Intuitionist type theory*) and GPT (*General problem theory* [15,17,18]). As we use GPT we will give only a brief explanation about it.

The description of a problem in predicate logic can be viewed within the goal of GPT, since it is able to describe the input and output data as well as the relation between them. It considers problems as mathematical structures, where solutions can be precisely treated and provides a framework for studying some problem-solving situations, as well as problem solving. However, these pieces of information aren't enough to assure the existence of a method that solves the problem.

Besides the specification in predicate logic and given that the sentence that describes a problem is a *theorem* of the specification, if we obtain a constructive proof we will be able to understand it not only as a syntactic transcription, but also as a description of a given object, in other words, a description of an algorithm [10].

The *Curry-Howard* (C.H.) isomorphism associates the inference rules in natural deduction for intuitionist logic (used in the proof) with the formation rules of λ-terms (commands in a programming language), in such a way that one proof of σ (a formula) can be seen as a λ-term, which has the type σ. Hence, we can say that a proof has a computational interpretation, that is, it can be seen as a set of commands in a programming language, i.e., a program [11]. This isomorphism gives the base for the construction process of the program from a proof that is generally called "*extraction of computational contents of a proof*"[2]. This process extracts a function that relates the input with the specific outputs of the program. The inputs and outputs of the program reflect the application of inference rules used to obtain the formula. The computational contents relate to the *semi-computational contents* that describe the relations between the inputs and outputs of the program. The input and output variables of the program, by the C.H. isomorphism are represented, respectively, by the variables quantified by the universal quantifier and existential quantifier, so, the theorem of the specification must be of form $\forall x \exists y\ P(x,y)$.

There are many proposals for constructive programs synthesis, which use constructive logic - for instance, the ITT- to specify the problems. These systems use as deductive system the sequent calculus ([4,3,9]) or the rewrite mechanism [6], and construct programs in logical and functional programming languages.

Based upon those considerations, this work proposes a constructive synthesizer, where the program is generated from a proof using natural deduction, avoiding the conversion that is used in the related work found in the literature. In this method, a program will be constructed in an imperative program language (Pascal-like) from a proof in many-sorted predicated intuitionist logic. Using the concept of semi-computational contents of a formula, we prove that the generated program is a true representation for the solution to the specified problem by the theorem of any many-sorted theory that describes the data types used by the problem.

In the next section, we will present our constructive synthesis program process, which is composed by the labeling of memory configurations of the program, followed by the association of each inference rule with commands in the imperative language. In the section 3, will be described the proof of correctness of the program synthesis, i.e., a proof that the generated program achieve the specification. Section 4 has an example of our constructive synthesis mechanism and finally, in the section 5, we will present the conclusion of the work.

---

[2] For more on this concept, see section 2

2 Program synthesis process

In the process of program synthesis we start from the existence of one theorem prover in many-sorted predicate intuitionist logic with arithmetic, which, beyond the usual inference rules, has inference rules for equality and induction. The theorem prover constructs a normal proof in natural deduction, for a certain theorem, which is the input to the programs synthesizer.

There are restrictions related to the inference rules used by the proof (given as an input to the synthesizer): 1- the proofs cannot have the negation introduction rule, 2- the existential elimination rule can be only applied on a formula that represents the inductive hypothesis. The last restriction[3] can be weakened if we admit parameterized programs as solutions.

From the proof of the specified problem we extract the computational content. In order to accomplish this, we first map all the memory configurations for each inference rule (labeling memory configuration process), and then we make the associate commands, in an imperative programming language, with each inference rule.

Labeling of memory configurations

We can view the execution of a program as a movement of bits inside the memory. Each movement represents an operation on the data of the program, which is stored in program's variables. Hence, it is very important to know the variables of the program and the values that may be attributed to them.

According to the operational reading of the connectives, given by the C.H. isomorphism, the variables quantified by universal quantifier are associated with the input data of the program and they are represented by the free variables since they can accept any value (of the same type of the variable) that make the formula true. The output data of the program are associated with the variables quantified by existential quantifier, which in a formula are represented by the free variables and the terms that depend on the input variables. Hence, the interpretation of the proof is based on the C.H. isomorphism, where each inference rule can be interpreted as a step in a program construction, the labeling of memory configurations process calculates the configuration of memory to each inference rule application.

The process of labeling memory configuration in a proof creates two sets: one for the input variables (free variables) and other to the output terms. In the beginning, both sets are empty. Next, the rules for labeling the input and output data will be used in the bottom-up direction. As the process reaches the proof tree leaves, we can find some variables and terms belonging to the set of free variables or to the set of output terms that will not be used as input and output of the program associated to the proof. These variables and terms reflect the memories data that are not used. They are considered residues of the labeling memory configuration process. These residues will be inserted in the set of input variables and output terms of the formulas that belong to the proof path derived from the formula, where they were detected for the first time (this process will be carried out in the top-down direction). Thus, the residues will be spread up to the proof tree root (conclusion) whose set of the input variables and the output terms will no longer be empty.

Labeling memory configurations rules

The labeling of memory configurations rules are related to the logical inference rules applications. In the presentation of the rules we will use the following notation: 1-α_T^V, where α is a formula, V the set of input variables and T the set of output terms; 2- $K \cup a$ represents the operation $K \cup \{a\}$, where K is either a list of input variables or the list of output terms.

---

[3] These restrictions will become clearer in the sequel.

The labeling rules below must be analyzed in the bottom-up direction, according to the labeling process[4].

Top-Formulae

axioms : β_T^V , where V and T are empty sets

Universal quantifier elimination:

$$\dfrac{\forall y\alpha(y)_T^V}{\alpha(a)_T^{V\cup a}}$$

Existential quantifier elimination

$$\dfrac{\exists y\alpha(y)_T^V \qquad \begin{array}{c}\alpha(a)_{T\cup a}^V\\ \vdots\\ \delta_{T'}^{V'}\end{array}}{\delta_{T'}^{V'}}$$

Conjunction elimination

$$\dfrac{(\alpha\wedge\beta)_T^V}{\alpha_T^V} \qquad \dfrac{(\alpha\wedge\beta)_T^V}{\beta_T^V}$$

Disjunction elimination

$$\dfrac{(\alpha\vee\beta)_{T'}^{V'} \qquad \begin{array}{c}[\alpha]_{T'}^{V'}\\ \vdots\\ \gamma_T^V\end{array} \qquad \begin{array}{c}[\beta]_{T'}^{V'}\\ \vdots\\ \gamma_T^V\end{array}}{\gamma_T^V}$$

Implication elimination

$$\dfrac{\alpha_{T'}^{V'} \quad (\alpha\rightarrow\beta)_T^V}{\beta_T^V}$$

Induction

$$\dfrac{\begin{array}{c}[k\prec l]_{T'}^{V'}\\ \vdots\\ \alpha(k)_T^{V\cup k}\end{array} \qquad \begin{array}{c}[\alpha(a_i)]_{T_1}^{V_1}\\ \vdots\\ \alpha(w)_T^{V\cup w}\end{array} \qquad a_i\prec w_{T_2}^{V_2}}{\forall y\alpha(y)_T^V}$$

Where k reflects the term associated to the base case and, w reflects the tem associated to the inductive case.

Hypothesis: β_T^V, where V and T contains the variables of the program associate with the hypothesis

Universal quantifier introduction

$$\dfrac{\alpha(a)_T^{V\cup a}}{\forall y\alpha(y)_T^V}$$

Existential quantifier introduction

$$\dfrac{\alpha(b)_{T\cup b}^V}{\exists y\alpha(y)_T^V}$$

Conjunction Introduction

$$\dfrac{\alpha_T^V \quad \beta_T^V}{(\alpha\wedge\beta)_T^V}$$

Disjunction introduction

$$\dfrac{\alpha_T^V}{(\alpha\vee\beta)_T^V} \qquad \dfrac{\beta_T^V}{(\alpha\vee\beta)_T^V}$$

Implication introduction

$$\dfrac{\begin{array}{c}[\alpha]_{T'}^{V'}\\ \vdots\\ \beta_T^V\end{array}}{(\alpha\rightarrow\beta)_T^V}$$

Intuitionist absurdity rule

$$\dfrac{\dfrac{\beta_{T_1}^{V_1} \quad \neg\beta_{T_2}^{V_2}}{\bot_{\{\}}^{\{\}}}}{\alpha_{\{\}}^{\{\}}}$$

Remark: In the labeling of memory configurations process, if a proof uses the equality congruency property, the resultant formulas of this rule utilization will have the set of the input variables and the output variables changed in the following way: the substituted variables or terms will be taken out from the set[5] where they belong, and they must be added to the terms and variables, on which the replaced terms depend, in their respective set[6].

Association of inference rules with instructions of an imperative language

In the program generation process each formula is related to a program that results from the association of inference rules with commands in a given programming language. Each association is based on the content (*logical* or *semi-computational*) of the formula, in such a way that a program will reflect the semantics of the proof of the formula associated.

---

[4] More details in [19].

[5] Output terms and input variables set.

[6] An example of the application of this rule in the section 4.

A formula has *logic content* when it is derived from an axiom or hypothesis and describes the nature of the objects used by the program to solve the problem proposed, i.e., it describes the data structures of a program and the set of operations that can be applied to them.

The *semi-computational content* of a formula is the set of information that express the relations between the input and the output data of a program – which is a solution for the problem specified by the formula - where for more than one input we can have one or more outputs.

The *computational content* of a formula is a function, within the semi-computational contents, which relates the input with the specific outputs of the program. They reflect the application of inference rules used to obtain the formula.

We use the following notation to describe the generation program rules: 1- $\Lambda:\alpha_T^V$ - where, Λ is a program that calculates the property described in α_T^V ; 2- Λ,Ψ,Γ - programs; 3- σ - description of the memory allocation; 4- p - name of the program related to the formula.
Remark: The commands of the language, in which the program will be produced, have the same semantics of the equivalent commands in the usual imperative programming languages.

Top-Formulae

Axioms : $\sigma:\beta_T^V$, where σ indicates "logical contents". **Hypothesis:** $\mathbf{p}:\delta_T^V$, where **p** is a symbol for programs

Universal quantifier elimination:

1.Axioms and non-inductive hypothesis: $\dfrac{\sigma:\forall y\alpha(y)_T^V}{\sigma:\alpha(a)_T^{V\cup a}}$ **2. Inductive hypothesis:** $\dfrac{p:\forall y\alpha(y)_T^V}{p:\alpha(a)_T^{V\cup a}}$

Universal quantifier introduction: $\dfrac{\Lambda:\alpha(a)_T^{V\cup a}}{read(a);\Lambda:\forall y\alpha(y)_T^V}$

Existential quantifier elimination:

$$\frac{\begin{array}{cc} & a\leftarrow exec(p,\bar{v}):[\alpha(a)^V_{T\cup a}] \\ p:\exists y\alpha(y)_T^V & \vdots \\ & \Gamma:\delta_T^{V'} \end{array}}{\Gamma:\delta_T^{V'}}$$

The command *exec*(...) gives to the program input variables the values passed by the parameters. Besides, it realizes the calling of the function and returns the last output term after the execution of ***p***.

Existential quantifier introduction: $\dfrac{\Lambda:\alpha(b)_T^{V\cup b}}{\Lambda;\, write\ \ (b):\exists y\alpha(y)_T^V}$

Conjunction elimination: $\dfrac{\Lambda:(\alpha\wedge\beta)_T^V}{\Lambda:\alpha_T^V}$ $\dfrac{\Lambda:(\alpha\wedge\beta)_T^V}{\Lambda:\beta_T^V}$

Conjunction Introduction: $\dfrac{\Lambda:\alpha_T^V \qquad \Psi:\beta_T^V}{\Lambda\otimes\Psi:(\alpha\wedge\beta)_T^V}$

Disjunction elimination:

$$\frac{\begin{array}{ccc} & \sigma:[\alpha]_T^{V'} & \sigma:[\beta]_T^{V'} \\ \sigma:(\alpha\vee\beta)_T^{V'} & \vdots & \vdots \\ & \Lambda:\gamma_T^V & \Psi:\gamma_T^V \end{array}}{if\ \ (\alpha)\ \ then\ \ (\Lambda)\ \ else\ \ (if\ \ (\beta)\ \ then\ \ (\Psi)):\gamma_T^V}$$

Disjunction introduction: $\dfrac{\Psi:\alpha_T^V}{\Psi:(\alpha\vee\beta)_T^V}$ $\dfrac{\Lambda:\beta_T^{V'}}{\Lambda:(\alpha\vee\beta)_T^{V'}}$

Implication elimination: The assertion associated to the implication elimination rule is: $\dfrac{\Psi:\alpha^{V}_{T}\ \Lambda:(\alpha\to\beta)^{V'}_{T'}}{[\Lambda\leftarrow\{exec(p,\vec{v})=\Psi\}]:(\alpha\to\beta)^{V'}_{T'}}$, where [Λ⇐{exec([p],v) = Ψ}], denotes the substitution of the supposed procedure call (p) by the real procedure call (Ψ) in the program(Λ). According to the proof restrictions (seen below), the minor premise of implication elimination rule always has logic contents. Thus, the program to be generated by the application of this rule is the program associated to the major premise. So, we have the following assertion: $\dfrac{\sigma:\alpha^{V}_{T}\quad \Lambda:(\alpha\to\beta)^{V'}_{T'}}{\Lambda:\beta^{V'}_{T'}}$

Implication introduction: The assertion related to this rule is: $\dfrac{\begin{array}{c} p:[\alpha]^{V'}_{T'} \\ \vdots \\ \Lambda:\beta^{V}_{T} \end{array}}{Dec\,p\ \ \Lambda:(\alpha\to\beta)^{V}_{T}}$, where *Dec* is a label to mark the utilization of procedure p. However, according to the proof's restrictions, the hypothesis used in the proof process has only logic content. Thus, we have the following assertion: $\dfrac{\begin{array}{c} \sigma:[\alpha]^{V'}_{T'} \\ \vdots \\ \Lambda:\beta^{V}_{T} \end{array}}{\Lambda:(\alpha\to\beta)^{V}_{T}}$

Intuitionist absurdity rule:

$$\dfrac{\begin{array}{c} \sigma:\beta^{V}_{T} \\ \vdots \\ \bot^{\{\}}_{\{\}} \end{array}}{\sigma:\gamma^{\{\}}_{\{\}}}$$

Induction: This rule is an alternative form of the application for the introduction of the universal quantifier. Consequently, the program generated will have the command related to the application of this rule (*read*(...)) and a recursive program formed by a conditional command

$$\begin{array}{ccc} [z\prec l\ ]^{V'}_{T'} & [p:\alpha(a_i)^{V_1}_{T_1}\ ] & \\ \vdots & \vdots & \vdots \\ \Psi:\alpha(z)^{V\cup z}_{T} & \Lambda:\alpha(k)^{V\cup k}_{T} & a_i\prec k^{V_2}_{T_2} \\ \hline \end{array}$$

```
Procedure Rec {   read(y);
                 if (y < l) then {Ψ}
                else                                  : ∀yα(y)^V_T
               { [Λ ← {exec([p], v̄) = Rec}*] }
              }
```

Observations: 1- If this rule will be the last rule to be applied on the proofs process, instead of ***Procedure***, this program will be declared as ***Program***; 2- In the assertion above, the * means the command related to the renaming of a hypothetical program ($\text{exec}([p],\bar{\text{v}})$) by the recursive call (**Rec**) in the program (Λ).

Remarks about introduction of implication rule: 1- the restriction that the minor premise always has logic contents is due to the fact that if it would have a program associated to it, the calling point in the program related to the major premise should be changed. As we are extracting programs from normal proofs, associated to the major premise, we only have the name of the program, in such a way that, we do not know in which place we have to change the procedure label by the procedure calling; 2- Due to the restrictions on the proof structure, the minor premise can be derived from the inductive hypothesis, which has a program associated. After the execution of the associated program we will have the configuration of memory for the execution of the program related to the major premise. Thus, we can interpret the program related to the minor premise as logic contents, satisfying the restriction of the implication elimination rule.

Restrictions about the rules used in the proof

The following rules will not be used because the extraction of their computational contents is not ordinary: **1- Negation Introduction rule-** This rule is directly related to the rule of the intuitionist absurdity, which according to the extraction of semi-computational contents, expresses that the allocation of memories is empty. So, we have empty memory allocation associated with the premise of the negation introduction rule. However there are examples that show that there is a semi-computational content for this rule, see[19]. Hence, this rule has to be better studied. **2- Existential quantifier elimination -** When we extract the semi-computational contents of a proof with this rule, we consider that there is a program associated to the major premise, which will be referenced through the label (*exec*) during the extraction process of the semi-computational contents (it will be changed by the program call). In the case of inductive hypothesis, the program related to the existential quantifier is the same program to be constructed. So, we will know what the program call that will substitute the label. Otherwise the program referred to is only hypothetical, and we will not know the shape of its call; thus, the user would be in charge of giving this information to the program.

3 Correctness proof of the synthesis process

The program synthesis process is composed of two parts: the labeling of memory configuration of the proof's formulas (*LabelMemoConfig*) and the extraction of computational contents of the proof (*GenProg*). With this method, we obtain a program that has the same semantics of the proof, and to guarantee it, we must to proof the correctness of the system, which is achieved by the proof of the theorem below:

$$\begin{tikzcd}
\dfrac{\Pi}{\forall x\exists y\alpha(x,y)} \arrow[r, Rightarrow, "LabelMemoConfig"] & \dfrac{\Pi}{\forall x\exists y\alpha(x,y)^{\{\}}_{\{\}}} \arrow[d, Rightarrow, "GenProg"] \\
 & \dfrac{\Pi}{\Lambda:\forall x\exists y\alpha(x,y)^{\{\}}_{\{\}}}
\end{tikzcd}$$

Figure2 – Schema of the program construction

Theorem: Let Π be a proof for a formula of the form $\forall x\exists y\alpha(x,y)$, from an axiom set (Δ) and a set of hypothesis that are not axioms (θ), and Λ the program provided by the function *GenProg*(*LabelMemoConfig* (Π)), then: $\theta,\Delta \models_{M,\sigma} \Lambda : \forall x\exists y\alpha(x,y)^{\{\}}_{\{\}}$

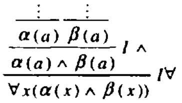
Proof: The proof of the theorem was carried out by induction in the length of the proof, through the comparison of the changed syntactic semantics – of the program – with the semi-computational contents of the inference rules.

4 Example

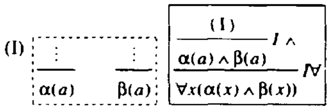
In this section we show our program synthesis mechanism through an example, in which a program that calculates the remainder of a division is generated. The proof tree will be presented in blocks. The block of main proof - which has the theorem to be proved - will be a solid frame. The others, with traced frames, represent the branches that are connected with others by the numeration presented in the upper left side of the frame.

Example:

Proof Tree:



Block Representation



To make easier the understanding of the proof, we use infix notation for addition, subtraction and multiplication operations, and the equality predicate and comparison predicates either. The functional s(x) expresses the operation of successor.

The program is generated from the theorem proof: $\forall v\forall u((v\to 0)\to\exists r\exists k(k*v=r=u)\wedge(r<v))$ on the basis of the axioms: $\forall x(x*1=x)$, $\forall x(x+0=x)$, $\forall q(0*q=0)$, $\forall z\forall p((z=p)\to(z-p=0))$, $\forall z\forall p((p>0)\to(z-p<z))$, $\forall z\forall q((z*s(q))\to(z*q+z))$, $\forall z\forall p\forall q((z=p-q)\to(z+q=p))$ and the hypothesis: l=y.

(I)

$$\dfrac{\dfrac{\dfrac{\sigma:\forall x(1*x=x)^{\{\}}_{\{1\}}}{\sigma:1*x=x^{\{x\}}_{\{1\}}}E\forall\quad\dfrac{\sigma:\forall h(h+0=h)^{\{\}}_{\{0\}}}{\sigma:x+0=x^{\{x\}}_{\{0\}}}E\forall}{\sigma:1*x+0=x^{\{x\}}_{\{0,1\}}}\quad\dfrac{\sigma:[b=l]^{\{b,l\}(1)}_{\{1\}}\quad\dfrac{\dfrac{\sigma:\forall z\forall p((z=p)\to(z-p=0))^{\{\}}_{\{0\}}}{\sigma:\forall p((b=p)\to(b-p=0))^{\{b\}}_{\{0\}}}E\forall}{\sigma:(b=l)\to(b-l=0)^{\{b,l\}}_{\{0\}}}E\forall}{\sigma:b-l=0^{\{b,l\}}_{\{0\}}}E\to}{\sigma:1*x+(b-l)=x^{\{x,b,l\}}_{\{(b-l),1\}}}Eq$$

(I)Continuation

$$\dfrac{\sigma:[x<0]^{\{x\}}_{\{0\}}\quad\dfrac{\sigma:[b=l]^{\{b,l\}(1)}_{\{1\}}\quad\dfrac{\dfrac{\sigma:\forall z\forall p((z=p)\to(z-p=0))^{\{\}}_{\{0,1\}}}{\sigma:\forall p((b=p)\to(b-p=0))^{\{b\}}_{\{0,1\}}}E\forall}{\sigma:(b=l)\to(b-l=0)^{\{b,l\}}_{\{0,1\}}}E\forall}{\sigma:b-l=0^{\{b,l\}}_{\{0,1\}}}E\to}{\sigma:b-l<x^{\{x,b,l\}}_{\{(b-l),1\}}}E\to$$

(II)

$$\dfrac{\dfrac{\sigma:\forall q(0*q=0)^{\{\}}_{\{0\}}}{\sigma:0*x=0^{\{x\}}_{\{0\}}}E\forall\quad\dfrac{\sigma:\forall h(0+h=h)^{\{\}}_{\{0\}}}{\sigma:0+b=b^{\{b\}}_{\{0\}}}E\forall}{\sigma:0*x+b=b^{\{x,b\}}_{\{b,0\}}}Eq\quad\dfrac{\sigma:[b<l]^{\{b,l\}(1)}_{\{\}}\quad\sigma:l=x^{\{x,b\}}_{\{l,0\}}}{\sigma:b<x^{\{x,b\}}_{\{b,0\}}}$$

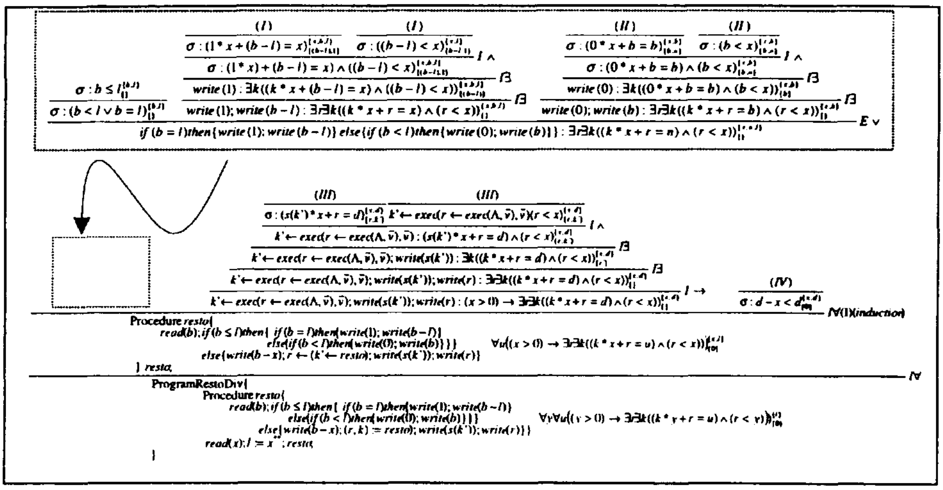
(III)

$$\dfrac{\dfrac{\dfrac{\Lambda:\exists r\exists k'((k'*x+r=d-x)\wedge(r<x))^{\{x,d\}}_{\{\}}\quad\text{(inductive hypothesis)}}{r\leftarrow exec(\Lambda,\vec{v}):\exists k'((k'*x+r=d-x)\wedge(r<x))^{\{x,d\}}_{\{r,k'\}}}E\exists}{k'\leftarrow exec(r\leftarrow exec(\Lambda,\vec{v}),\vec{v}):((k'*x+r=d-x)\wedge(r<x))^{\{x,d\}}_{\{r,k'\}}}E\exists}{k'\leftarrow exec(r\leftarrow exec(\Lambda,\vec{v}),\vec{v}):(k'*x+r=d-x)^{\{x,d\}}_{\{r,k'\}}\;k'\leftarrow exec(r\leftarrow exec(\Lambda,\vec{v}),\vec{v}):(r<x)^{\{x,d\}\,***}_{\{r,k'\}}}E\wedge$$

$$\dfrac{\dfrac{k'\leftarrow exec(r\leftarrow exec(\Lambda,\vec{v}),\vec{v}):(k'*x+r=d-x)^{\{x,d\}}_{\{r,k'\}}\quad\dfrac{\dfrac{\dfrac{\sigma:\forall z\forall p\forall q((z=p-q)\to(z+q=p))^{\{\}}_{\{\}}}{\sigma:\forall p\forall q((k'*x+r=p-q)\to(k'*x+r+q=p))^{\{x\}}_{\{\}}}E\forall}{\sigma:\forall q((k'*x+r=d-q)\to(k'*x+r+q=d))^{\{x,d\}}_{\{\}}}E\forall}{\sigma:(k'*x+r=d-x)\to(k'*x+r+x=d)^{\{x,d\}}_{\{\}}}E\forall}{\sigma:(k'*x+x+r=d)^{\{x,d\}}_{\{r,k'\}}}E\to\quad\dfrac{\dfrac{\sigma:\forall z\forall q(s(q)*z=q*z+z)^{\{\}}_{\{\}}}{\sigma:\forall q(s(q)*x=q*x+x)^{\{x\}}_{\{\}}}E\forall}{\sigma:(s(k')*x=k'*x+x)^{\{x\}}_{\{k'\}}}E\forall}{\sigma:(s(k')*x+r=d)^{\{x,d\}}_{\{r,k'\}}}Eq$$

$$\dfrac{k'\leftarrow exec(r\leftarrow exec(\Lambda,\vec{v}),\vec{v}):(r<x)^{\{x,d\}\,***}_{\{r,k'\}}}{k'\leftarrow exec(r\leftarrow exec(\Lambda,\vec{v}),\vec{v}):r<x^{\{x,d\}}_{\{r,k'\}}}$$

$$\dfrac{\sigma:[x>0]^{\{x\}}_{\{0\}}\quad\dfrac{\dfrac{\sigma:\forall z\forall p((p>0)\to(z-p<z))^{\{\}}_{\{0\}}}{\sigma:\forall p((p>0)\to(d-p<d))^{\{d\}}_{\{0\}}}E\forall}{\sigma:(x>0)\to(d-x<d)^{\{x,d\}}_{\{0\}}}E\forall}{\sigma:d-x<d^{\{x,d\}}_{\{0\}}}$$



$**$In the proof hypothesis $y=l$ represents a memory restriction, where l has the value of y.

Remark: In our system the communications between the functions (parameter passing) is made by read and write file operations.

5 Conclusion

In this work we have presented an automatic method of program synthesis operating as follows: it transforms a given proof based on a specification to a program (in a imperative language), guaranteeing the correctness of the latter.

The syntactical restrictions imposed on the proof, from which we extract the semi-computational contents may cause some loss of the expressive power, thus limiting the domain of application of the synthesis procedure.

Among the main contributions of this work, we stress the proposal of a new synthesizer that generates legible programs in an imperative language along with a correctness proof of this mapping. The other synthesizers exposed in the existing literature generate programs that are not very legible in functional or logical programming languages. Also, our constructive program synthesis procedure receives as input a declarative specification in predicate logic, which allows us to express the problem in a simple way than the synthesizers using intuitionistic type theory (**Nuprl**[4], **Oyster**[3] e **NJL**[9]) and equational logic (**Lemma** [6]).

Among the directions to extend this work, we expect to investigate the feasibility of relaxing some restrictions on the proofs, so as to extract semi-computational contents from proofs that use the negation, introduction rule or have existentially quantified formulas as hypothesis. Also, the usage of more that one proof method seems attractive and program synthesizers based on such ideas are being investigated.

References

[1] BENL, H., BEGER, U., SCHWICHTENBERG, H., SEISENBERGER, M. and ZUBER,W. Proof theory at work: Program development in the Minlog system, *Automated Deduction, W. Bibeland P.H.Schmitt,*(eds.), Vol II, Kluwer 1998

[2] BATES, J.L. and CONSTABLE, R.L.- "Proof as Programs". *ACM Transactions on Programming Languagens and Systems,* 7(1): 113-136,1985.

[3] BUNDY, A., SMAIL, A., and WIGGINS, G. A. - "The synthesis of logic programs from inductive proofs". In J. Lloyd (ed.), *Computational Logic,* pp 135-149. Springer-Verlag,1990.

[4] CALDWELL, J.L., IAN, P., UNDERWOOD, J.G. - "Search algorithms in Type Theory". http://meru.cs.uwyo.edu/~jlc/papers.html

[5] CHANG, C., LEE, R.C - "*Symbolic Logic and Mechanical Theorem Prover*". Academic Press, 1973

[6] CHARARAIN, J. e MULLER, S. - "Automated synthesis of recursive programs from a ∀∃ logical Specification". *Journal of Automated Reasoning* 21: 233-275, 1998.

[7] DEVILLE, Y., LAU, K. - "Logic Program Synthesis"- *Journal of Logic Programming* 1993:12:1-199

[8] FLOYD, R. - "Assigning meaning to programs". *Symposia in Applied Mathematics* 19:19-32, 1967.

[9] GOTO, S. - "Program synthesis from natural deductions proofs". *International Joint Conference on Artificial Intelligence,* 339-341. Tokyo 1979.

[10] GIRARD, J., LAFONT, Y. and TAYLOR, P. - *Proof and Types.* Cambrigde University Press, 1989.

[11] HOWARD, W.A. - "The Formulae-as-Types Notion of Construction". In Hindley, J.R., Seldin, J.P.(ed.), *To H.B. Curry: Essays on combinatory logic, Lambda Calculus and Formalisation.* Academic Press, 1980.

[12] HOARE, C.A.R and WHIRTH, N. - "An axiomatic Definition of the Programming Language PASCAL" -December, 1972. *Acta Informatica* 2: 335-355, Springer-Verlag, 1973.

[13] KREITZ, C. - "Program synthesis - Automated Deduction - A basis for Applications", pp 105-134, Kluwer,1998.

[14] LAU, K. and WIGGINS, G. - "A tutorial on Synthesis of Logic Programs form Specifications". In P. Van Hentenryck (ed.), *Proceedings of the Eleventh International Conference on Logic Programming,* pp 11-14, MIT Press, 1994.

[15] MARTIN-LÖF, P. - "Intuitionistic Type Theory".*Edizioni di filosofia e Scienza,* Biblioplolis, 1984.

[16] MANNA,Z. and WALDINGER,R. - "A Deductive Approach to program synthesis". *ACM transactions on Programming Languages and Systems,* 2(1):90-121, 1980

[17] VELOSO, P.A.S. - "Outlines of a mathematical theory of general problems". *Philosophia Naturalis,* 21(2/4): 234-362, 1984.

[18] HAEUSLER, E.H. - "Extracting Solution from Constructive Proofs: Towards a Programming Methodology". *Brasilian Eletronic Journal on Mathematics of Computation* (BEJMC). No. 0- Vol 0, 1999. (http://gmc.ucpel.tche.br/bejmc)

[19] SILVA, G.M.H. - "Um Estudo em Síntese Construtiva de Programas utilizando Lógica Intuicionista". Dissertação de Mestrado – Departamento de Informática -PUC-Rio. 1999.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

An Agent-Oriented Inference Engine applied for Supervisory Control of Automated Manufacturing Systems

Jean Marcelo Simão, Paulo Cézar Stadzisz

Centro Federal de Educação Tecnológica do Paraná
Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial,
Av. Sete de Setembro, 3165, 80230-901, Curitiba, PR, Brazil
simao@cpgei.cefetpr.brs, stadzisz@cpgei.cefetpr.br

Abstract: This paper proposes an agent-based architecture to implement expert systems. The reactive and co-operative agents encompass the facts base and the set of rules. A *Rule* is composed by a *Condition* (with connected *Premises*) and an *Action* (with connected *Orders*). The facts with a strong cohesion are aggregated in an *Attribute*, becoming its states. To change a state in Attribute is used a Method. When a set of *Attributes* and Methods are supervising the different aspects of some object, this set is encapsulated in a *Fact Base Agent*. The *Premises* evaluate *Attributes* and *Orders* activate *Methods*, both can be shared by *Rules* avoiding redundancies. Each *Attribute*, knowing what *Premises* has interest in their states change, straightforward notifies them when a fact occurs. Then the *Notified Premises* make a new logical calculus and, if there are one or more changes, they notify the *Interested Rules*, which recalculate their new boolean values. The notification principle by agents is a significative contribution in the Inference Engine accomplishment, eliminating search and bringing a quicker response to the changes. It is also proposed agents to make easier the creation of the rules. As an application instance, an architecture for the design and implementation of Supervisory Control Systems for Automated Manufacturing is derived from the proposed approach. The tests of the architecture are done using a robust simulation tool called ANALYTICE II. The main contribution of this work is an archetype to production systems carried out by agents and implementing an advanced inference.

1. INTRODUCTION

Expert systems (ESs) have been successfully applied in many areas (e.g. medical, games and engineering). The use of expert human knowledge in computational systems has motivated an increasing number of researches in ES domain [7][10].

Basically, an ES consists of facts about the observed system, causal relations (rules) which specifies the different ways to change the facts, and an inference engine (IE) to effectiveness changes them. The IE has an inference cycle, which is the process to match rules and facts to obtain the proved rules, when a new fact happens [7][10].

There are some difficulties in the development of ESs, as the construction of efficient IE with acceptable response time. If the goal is to design a small-scale ES, it would be trivial to develop a simple program where in each inference cycle all facts would be match with all rules. However, this approach results in a computational complexity that is exponential with respect to the number of rules [7][9][10].

It was needed more advanced policies to build IEs, like the very used Rete networks proposed by Forgy [5]. The kernel of many commercial expert systems (e.g. ART, CLIPS, OPS83 and ILOG Rules), designed to handle hundreds or thousands of rules, use this technique or a variant thereof [9].

This paper proposes a new approach to implement ES, where agents represent both facts and rules, and the IE is reached by their relationship. The agents applied in this work are reactive and co-operatives, following the relations established by an object-oriented architecture, which can be considered as a multi-agent approach [1][8][11][14]. The main improvements are the search elimination in the inference process by using a notification mechanism and the functional independence in the architecture composition obtained by encapsulating and sharing the rule components and elements of facts base.

As instance of application, a Rule Based Systems (RBS), derived from the proposed architecture, is applied to solve the problem of Supervisory Control of an Automated Manufacturing System (SCAMS). SCAMS is a discrete control responsible for co-ordinating the factory components (e.g. lathes and robots) carrying out predefined production process [2][4][12]. The design of a SCAMS is a complex activity due to the automation level, required flexibility and involved risks, costs and time. Therefore, it is an appropriate example to demonstrate the robustness of the proposed approach.

This article is organised as follows: Section 2 describes expert systems and production systems, Section 4 presents the proposed architecture, Section 5 presents the architecture for SCAMS and Section 6 presents the conclusions of this work.

2. PRODUCTIONS SYSTEMS AND EXPERT SYSTEMS

An ES normally works in an application domain with substantial knowledge, from expert humans, stored in a base of facts. The ES need some reasoning mechanism to reach its objectives. It also may need a way to present its conclusions to the user and a way to keep actualised its inference base (i.e. rules).

The most frequent practice for knowledge representation in ESs are the production rules, making ESs be currently referenced to as RBS. Rules are connected to a frame system that defines the objects that are referenced in the rules. A frame is a representation of a complex element; it is identified by a *name* and consists of a set of slots. In some ways, the Frame approach is similar to the Object-Oriented approach [7][10][11].

A lot of large-scale ES, such as the CLIPS system from NASA, are founded on the concept of production systems (PS). A PS is composed by a **working memory (WM)** for data (or facts); a **production memory (PM)** for rules (where left-hand side is the *condition* and the right-hand side is the *action*); and an **inference engine (IE)** that infers new facts from existing facts, making the match with the rules, and asserts the new facts into the WM. The matching can be forward (i.e. to search the conclusion) or backward (i.e. to search the condition). The new facts can allow new inferences, and this cycle will finish when no further facts can be inferred [7][9].

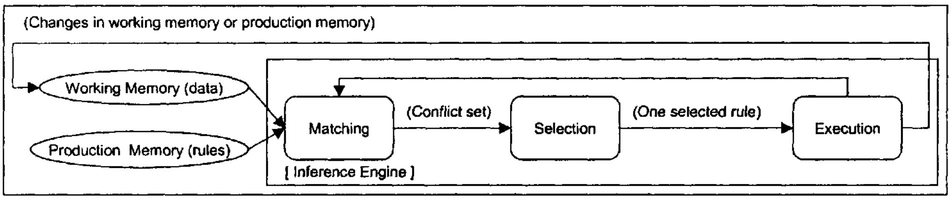


Figure 1- The Cycle Operation of Inference Engine in a Block Diagram.

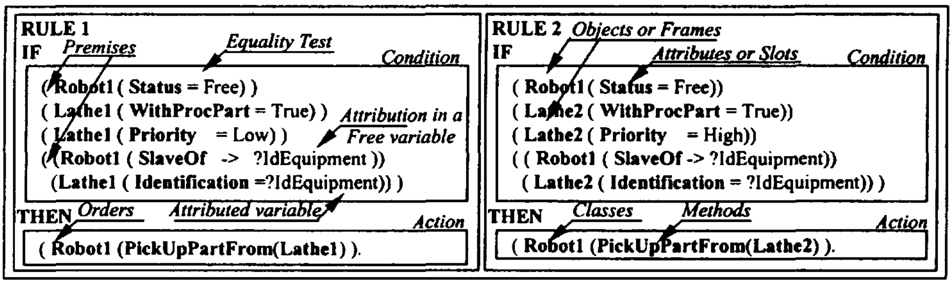
An ES can be considered as a PS due to the operation cycle of its IE. This cycle consists of three steps called *matching*, *selection* and *execution*, as demonstrated in Figure 1. The *matching* step propagates the latest facts to the WM and finds which rules are enabled, determining a conflict set (also called *agenda*). The *selection* step does the ordering of the rules in the agenda, using some parameters (e.g. the last date in which the rule was fired, the recency of facts from the WM or the simplicity/complexity of the rule) and the rules with better punctuation are, in order, selected from the agenda. The *execution* carries out the actions of the selected rules. These actions may either assert the new inferred facts into the WM or invoke external functions to the PS. These features are the nutshell of a PS [9][10].

3. PRODUCTION SYSTEM CARRIES OUT BY AGENTS

The PS concept is used in a large number of works and applications including the work of Forgy, called Rete network [5]. Rete is a reference in advanced IE and it has been used in traditional ESs to improve the computational capacity of matching rules and facts [9][10].

To present the approach propose in this paper and related concepts, the following two rules (in Figure 2) will be considered. These rules were developed to control a hypothetical manufacturing cell where the facts are frames.

In the Rule 1, the components of the rule are indicated. This work considers that the *Condition* is concerned with *Premises* and the *Action* is concerned with *Orders*. Each Premise that compares the value of an attribute with a constant is called a *Simple Premise* (SP), and each Premise that compares values of attributes is defined as *Composed Premise* (CP). In the Rule 2, the composition of *Premises* and *Orders* is indicated for both object-oriented and frame point of view.



```
RULE 1
IF
 ( Robot1 ( Status = Free) )
 ( Lathe1 ( WithProcPart = True) )
 ( Lathe1 ( Priority    = Low) )
 ( (Robot1 ( SlaveOf  ->  ?IdEquipment ))
  (Lathe1 ( Identification =?IdEquipment)) )
THEN
 ( Robot1 (PickUpPartFrom(Lathe1) ).
```

```
RULE 2
IF
 ( Robot1 ( Status = Free))
 ( Lathe2 ( WithProcPart = True))
 ( Lathe2 ( Priority    = High))
 ( ( Robot1 ( SlaveOf -> ?IdEquipment))
  ( Lathe2 ( Identification = ?IdEquipment)) )
THEN
 ( Robot1 (PickUpPartFrom(Lathe2) ).
```

Figure 2 – Rules and its constituents.

In IE it is important to eliminate the temporal redundancies in the matching process, i.e. computations for matching the facts with the *Premises* should be carried out once. It is also important the elimination of structural redundancies, i.e. to share *Premises* between different *Conditions* of *Rules*. Without these two redundancies, it is possible to reduce the computational complexity, related to the number of rules, to polynomial time complexity [5][9]

A Rete network, for example, eliminates the temporal redundancy. But the structural redundancy is eliminated only for SPs, not for CPs. Nevertheless, it implements an optimised search process, putting all the components of rules in a tree-structured sorting network used to feature tests.

This paper proposes a different paradigm to create and carry out PS, where there is no search in inference process. Instead, a robust solution called **notification** is used. This approach eliminates temporal and structural redundancies.

The proposed solution is expressed in a generic computational archetype. This architecture is object-oriented, where both facts and rules are computational small objects, currently called agents [1][6][8][11][14]. The *facts base* is self-contained, the *rules* are reactive and highly connected in a graph, which allows they co-operate accomplishing the inference process.

An agent can be defined as a software module, with high degree of cohesion, with well-defined scope, with autonomy and taking part of a certain context whose changes are perceived by the agent. These perceptions may change the agent behaviour and it may promote another changes in the context [1][6][14].

3.1 Base of Facts or Working Memory

Facts extremely correlated (e.g. as those taking boolean values or representing states with mutual exclusion) are encapsulated and treated in an **Attribute-Agent (AT)**. This agent has a special feature: it knows what *Premises* are "interested" in its facts occurrence. In other s words, the agent knows what Premises must be notified of changes in its state (value). The values of an AT define its discrete states.

When a set of ATs makes reference to a same entity in the observed system, the ATs are aggregated into a **Fact Base Agent (FBA)** becoming its attributes. As example, if a nuclear plant is over ES supervision, all correlates ATs to a specific reactor (e.g. pressure or temperature ATs) could be included in a specific FBA to observe it, whose general state is given by the values (states) of its ATs.

The AT's states are the facts observed by the *Rules*. When a *Condition* of a *Rule* is proved, the respective *Action* can be fired. The related *Orders* may instigate changes in the ATs states. However, the AT need another FBA agent, called **Method-Agent (MA)**, to carry out the changes.

3.2 Rules

A rule is computationally represented by a **Rule Agent (RA)**, composed of a **Condition Agent (CA)** and an **Action Agent (AA)**, which have a causal relation. CA and AA represent, respectively, the *Condition* and the *Action* of the rule.

A **CA** carries out the logical calculus for the RA by the conjunction of boolean values from the connected **Premise Agents (PAs)**. There are simple and compose PAs respectively representing the SPs and CPs. A simple **PA** points to an AT from a FBA (the ***Reference***), has a logical operator (the ***Operator***) and has a limit of values (the ***Value***). This PA obtains its boolean value comparing the *Value* and *Reference* using the *Operator*. The composed PA still can has the *Value* as a pointer to another AT. Others CAs may share whichever PA.

An **AA** is connected to **Orders Agents (OAs)**, which represent the *Orders* of a rule. The OAs change states of FBAs (by means of their MAs) referenced in the respective CA. Each AA is linked to a CA and will only be prone to execution if the evaluation produced by this CA results true.

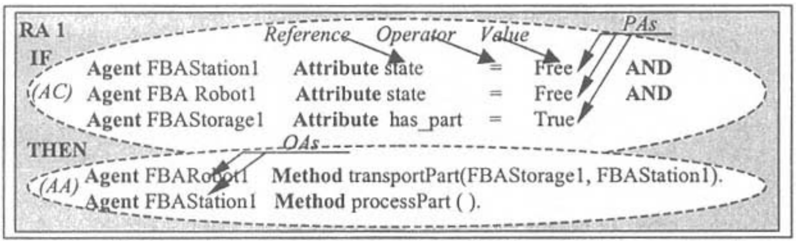

RA 1
IF
(AC)
Reference
Operator
Value
PAs
Agent FBAStation1 Attribute state = Free AND
Agent FBA Robot1 Attribute state = Free AND
Agent FBAStorage1 Attribute has_part = True
THEN
OAs
(AA)
Agent FBARobot1 Method transportPart(FBAStorage1, FBAStation1).
Agent FBAStation1 Method processPart ( ).

Figure 3 – An instance of Agent Rule.

The Figure 3 demonstrates a rule in the form of agents to control a manufacturing cell. This rule analysis if the Station1 is Free, if the Robot1 is also Free and if the Storage1 has a part.

3.3 Inference Process by Notification

Whenever a change in an AT value occurs, it notifies the interested PAs that reset their logical calculus. If a state change occurs in some PA, it notifies the CAs interested, since each PA knows which CAs are connected. These CAs reappraise then its own logical value. The 'true' value in a CA proves the respective RA, making the AA execution possible.

The IE is carried out by inter-relation of agents using a connection graph. In this solution the inference is quicker, because there are not list or tree searches, but messages propagation from ATs, through PAs and RAs, reaching AAs when pertinent. Figure 4 illustrates, using an Activity Diagram (UML notation) [11], the notification chaining after the occurrence of state changes in FBAs.

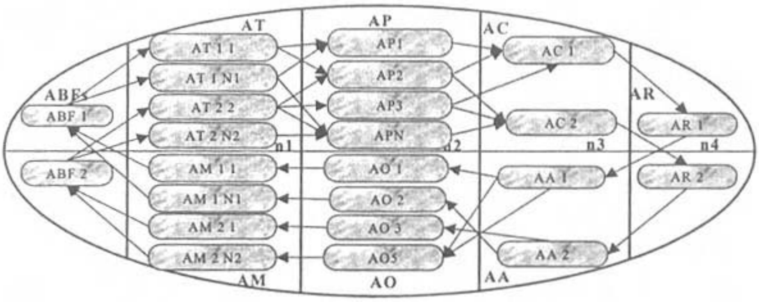


Figure 4 - The notification mechanism carried out by agents' co-operation.

This notification mechanism allows guarantying the temporal independence, since a new inference takes place only when a new fact occurs and only the *Premises* affected are re-evaluated. All the previous logical calculus remain valid.

3.4 Conflict among Rule Agents

A conflict occurs when a PA has as *Reference* an AT of an *exclusive FBA* and this PA is being shared among RAs with true boolean values (eligible RAs). An *exclusive FBA* is a FBA representing an exclusive resource, i.e. a resource that can be only used once a time. The AT can be considered by only one rule, because after the fire of this rule, the fact will be changed and will not be true anymore for the other rules.

To solve the conflict, it is chosen only one RA to be activated (the elected RA). This is done based on decision parameters that may arise from many origins (depending of the SBR application). In proposed approach, this decision is encapsulated in the Solver Agent.

A conflict instance could be met in the *Premise* '**Agent** FBARobot1 **Attribute** state = Free' (Figure 3) which may collaborate to turn true two ARs. The fact "Free" in FBARobot1 is an *exclusive fact*. The robot in this state could serve exclusively one of the workstations by proving one of the AR. Consequently, only one of the ARs can be executed, creating a conflict. In the context of control systems, a Solver Agent could be constructed including a number of decision parameters (e.g. scheduling policies).

3.5 Formation Rules and Agents

In the previous example, rules were composed using instance-oriented Premises. Each *Premise* makes reference to one or two objects (i.e. ATs). Then, the rule composition could become difficult due to the specialization level of the premises and actions.

To simplify the creation of rules, it is proposed the *Rule* conception by means of class-oriented rules, called **Formation Rules (FRs)**. A FR, as illustrated in Figure 5, is more generic than a *Rule*, therefore its premises only analyse if a FBA is from a given class (e.g. class A or B) without considering, at first, the ATs values. A FR filter creates combinations of agents from the facts base, where each agent pertains to one derived class in the specific architecture application. Each combination creates a *Rule* following the structure specified by the FR elements. A FR is computationally represented by a Formation Agent (FA).

Each FA has a **Condition Formation Agent (CFA)** and an **Action Formation Agent (AFA)**. The **CFA** is connected to **Premises Formation Agents (PFAs)**. The main function of the PFA is to filter agents pertaining to a given class. Each PFA specifies a ***Reference*** (to an AT of a class), a ***Value*** and an ***Operator***. This information is not used at first, but may be used as a model for the creation of PAs. The role of a CFA is to find the combinations among the PFAs filtered elements for its FA. The **FAA** is a sequence of **Orders Formation Agents (OFA)** that serves only as models for the creation of OAs.

Each resulting FA combination allows a RA instancing. The associates of the FA (i.e. CFA, PFA, AFAs and OFAs) create the RA's associates (respectively CA, PAs, AA and OAs). After the creation of a CA, the necessary PAs are created and connected. At the PA creation, it receives the *Reference*, the *Value* and the *Operator*, in accordance to the FA specifications. After the creation, the PA executes its logical calculus and attributes to itself a boolean value. After having its PAs connected, each RA may know its boolean value. As more RAs are created and need already existent PAs, connections between those PAs and the RAs are done, avoiding redundancies and generating a graph of connections. The AA, after being created, is connected to a sequence of OAs, in accordance to the FA's specifications. The information of an AFA is used only to create the AAs.

The PFAs will just analyse the restrictions of the FBAs attributes if it has been explicitly determined. This analysis is the *earlier logical calculus*, where a FBA is filtered only if the boolean result is true. In summary, this resource is a pruning in the graph of combination avoiding the creation of RAs without semantic meaning (i.e. RAs that would never reach a 'true' state).

FRs are means to assist the Rule creation process. The architecture allows the direct creation of Rules, without the use of FRs. As well as Rules, FRs need a way (maybe an agent) to extract the knowledge in the respective FAs and Ras. FAs, like RAs, still have the ability to share elements (i.e. PFAs and OFAs), avoiding redundancies in the filtering process.

Figure 5 exhibits a FR in the form of agents to create control rules for a manufacturing cell. It filters combinations of agents from the facts base, where each agent pertains to one of following classes: FBAtation, FBARobot and FBAStorage.

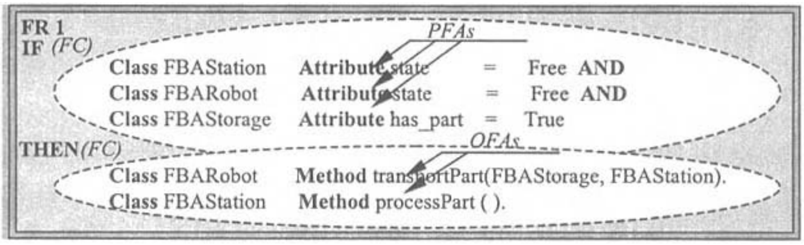

FR 1
IF (FC)
PFAs
Class FBAStation Attribute state = Free AND
Class FBARobot Attribute state = Free AND
Class FBAStorage Attribute has_part = True
THEN (FC)
OFAs
Class FBARobot Method transportPart(FBAStorage, FBAStation).
Class FBAStation Method processPart ( ).

Figure 5 – An instance of Agent Formation.

4. RULE AND AGENT BASED SUPERVISORY CONTROL ARCHITECTURE

As instance, the proposed approach was applied to SCAMS, an important area in control engineering. A SCAMS is dynamic, event-oriented system whose controlled variables are changed when discrete events occur. It is integrated with other functions in a plant as Planning, Scheduling and Supervision to make the production process management [2][4].

The SCAMS is responsible for co-ordinating the factory components (e.g. lathes and robots) to carry out predefined production process. This co-ordination includes: monitoring of the discrete states from each element in the factory (i.e. facts); decision about these states and current process (i.e. condition); and actuation over factory components by means of appropriate commands and respecting specific protocols (i.e. action) [2][12].

In this example, the ATs understand and explicit the discrete events (e.g. *robot is free*, *storage has a part* and *lathe is occupied*) and MEs command the components (e.g. to move a robot's end-effector). To better represent, monitor and command, the FBAs are specialised in **Active Command Components** (ACC) for physical components (e.g. machines and parts), **Control Unity** (CU) for the hierarchy (e.g. station and cell) or **Abstract Agent** (AB) for abstract elements that come from other decision elements (e.g. process plan and lot of parts from scheduler).

The *Rule* structure (i.e. RAs) is essentially the same as in base architecture, since their function is to analyse information from a standard interface (i.e. AT and MAs). FRs (i.e. FAs) are also the same, but now working over specialised FBAs [12][13].

The derived architecture was experimented over an advanced AMS simulator, called ANALYTICE II, which allows reproduce the mains factory features. SCSAM does not interact with real but with simulated equipments. Tests demonstrate the easiness for the creation of the SCSAM and the robustness of the constituted control system [13].

The control of discrete event systems has been extensively studied in academy and in industry research centres. Many approaches and methods for design and implementation have been proposed [3][4]. The developed SCSAM using the notification, agent-based approach described in this paper, is an innovation in this research area and can be viewed as a very flexible and efficient model to implement supervisory control for industrial plants.

5. CONCLUSIONS

The paper presented a new approach to create and carry out ES, where advances are met by the use of reactive and co-operative agents. Agents are designed and implemented using an object-oriented paradigm, consisting of a software architecture with advantageous characteristics, like functional independence, easy comprehension and derivation, trade-off between generality-applicability and robust execution of instances.

In this generic PS, the implementation of a notification mechanism was possible due to the WM (facts) and the PM (rules) be self-contained agents, with some autonomy, and distributed in a graph of connections. With this mechanism, searches are unnecessary and temporal redundancies are eliminated. Logical calculus only occurs when there is a new fact that is propagated only over the related elements. Notification mechanism makes possible an incremental inference by its straightforward notifications, bringing to a quicker response to the changes. Another feature of this approach is the sharing of *Premises* between *Rules*, eliminating the structural redundancies and also improving the inference process

The architecture still facilities the conception of *Rules*, allowing to its construction in a generic way, by means of class-oriented *Premises* and agents that generates specific *Rules* with object-oriented *Premises*.

This generic approach specifies the base of fact in a standard form. For create a derived architecture, it is necessary that the facts be encapsulated in AT and the change of facts in MAs. The architecture was specialised in RBS to SCSAM and, in fact, the main derivation occurs in the FBA, in this case to treat different elements of factory. The tests over

ANALYTICE II demonstrate that the principles of architecture are actually functional and derived architecture is robust and applicable for different instance of SCSAM.

The notification principle, carried out by agents, to implement an IE is the main feature of the proposed approach. The architecture may be applied in a number of applications, including supervisory control systems. Future works include the expansion of the applications and the development of a framework, including distributed processes and tools, to assist the fact base and rule module creation. Others works include the development of methods for the causal relation synthesis using Petri Nets and mapping techniques from these solutions to implementations under the proposed architecture.

References

[1] Ávila, B. C. & Prado, J. P. de A. & Abe, J. M., 1998: "Inteligência artificial distribuída; aspectos". Série Lógica e Teoria da Ciência, Instituto de Estudos Avançados – Universidade de São Paulo.

[2] Bongaerts, L., 1998: Integration of Scheduling and Control in Holonic Manufacturing Systems. Ph. D. Thesis PMA / Katholieke Universiteit Leuven.

[3] Chaar, J. K. & Teichroew, D. & Volz, R. A., 1993: Developing Manufacturing Control *Software*: A Survey and Critique. The International Journal of Flexible Manufacturing Systems. Kluwer Academic Publishers. Manufactured in The Netherlands, pp. 53-88.

[4] Cury, J. E. R., de Queiroz, M. H. e Santos, E. A. P., 2001. Síntese Modular do Controle Supervisório em Diagrama Escada para uma Célula de Manufatura. V Simpósio Brasileiro de Automação Inteligente, Canela, RS, Brasil.

[5] Forgy, C. L., 1982: RETE: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. Artificial Intelligence.

[6] Franklin, S. & Graesser, A., 1996: Is it an Agent, or Just a Program? A Taxonomy for Autonomous Agents, Institute for Intelligent Systems – University of Memphis, Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages, Springer-Verlag.

[7] Jackson, P., 1990: Introduction to Expert Systems. Addison-Wesley.

[8] Müller, J. P. 1998: Architectures and Applications of Intelligent Agents: A Survey. International House. Ealing London W5 5DB. Knowledge Engineering Review.

[9] Pan, J. & DeSouza, G. N. & Kak, A. C., 1998: "FuzyyShell: A Large-Scale Expert System Shell Using Fuzzy Logic for Uncertainty Reasoning. IEEE Transactions on Fuzzy Systems, Vol. 6, No 4, November.

[10] Rich, E. & Knight, K., 1991: Artificial Intelligence. McGraw-Hill.

[11] Rumbaugh, J. & Jacobson, I. & Booch, G., 1999: The Unified Modeling Language Reference Manual. Ed. Addison Wesley Longman.

[12] Simão, J. M., 2001: Proposta de uma arquitetura para sistemas flexíveis de manufatura baseada em regras e agentes. Master of Science Thesis. CPGEI, CEFET-PR. Brasil.

[13] Simão, J. M. & Silva, P. R. O. da & Stadzisz, P.C. & Künzle, L. A., 2001: Rule and Agent Oriented Software Architecture for Controlling Automated Manufacturing Systems. Pg 224 in Logic, Artificial Intelligence and Robotic – Frontiers in Artificial Intelligence (Serie) – LAPTEC. IOSPress Ohmsha.

[14] Yufeng, L. and Shuzhen, Y., 1999: Research on the Multi-Agent Model of Autonomous Distributed Control System, In 31 International Conference Technology of Object-Oriented Language and Systems, IEEE Press, China.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

LTLAS: a Language Based on Temporal Logic for Agents Systems Specification

NICANDRO FARIAS MENDOZA and FELIX F. RAMOS CORCHADO
Electric Engineering Department of CINVESTAV-IPN Guadalajara
Guadalajara, Jalisco telephone 31345570, fax 31345579,
e-mail {nmendoza, framos}@gdl.cinvestav.mx

Abstract

The multiagent systems paradigm represents one of the most promising approaches to the development complex systems. However, multiagent system specification is a hard task, overcoat if it is necessitated a formal specification. In this paper we develop a language we call LTLAS equivalent to the AML language developed by M. Wooldridge. AML takes as theoretical models, the intentional approach to agent-oriented paradigm and the temporal logic to construct a formal model. As AML, our language LTLAS has the power to make a formal specification but is easier to use and understand because it looks like any programming language commonly used by developers. Thus, the main contribution of this paper is the creation of the object language based on temporal logic, which let us give a computational interpretation for AML rules.

Key words: Agents, Temporal Logic, Agent Language, and Formal Specification

1. Introduction

The development of distributed applications comes to be a complex problem. The main cause is that distributed applications need coordination, cooperation and interaction process among the different distributed entities. The multiagent systems represent one of the most promising approaches to solve these problems. However, as any other paradigm this approach necessitates formal specification techniques to ensure the match among specification and implementation. The main problem using formal specification to develop systems, is the complexity faced by users, which ought to deal with formulas of temporal logic [], algebraic equations [] or Petri Networks []. This problem motivates us to develop a language with syntax more easy to understand but with power necessary to specify formally agent-based systems.

The works we consider important for our article are those of: M. Wooldridge [16] who develop a BDI logic used to establish the agents metalanguage AML, Shoham [12] who proposed an agent-oriented language called AGENT0, R. Thomas [15] she proposed the PLACA agent programming language, and N. Farías [3] who proposed the LCIASA Agent Communication Language.

Today, the predominant approach to express MultiAgent Systems consists on treating them as intentional systems, which can be understood attributing them mental states, such as desires, beliefs and intentions. The most successful approach to deal with intentional systems is with the temporary logic as is shown by important works as: the theory of intention of Cohen–Levesque [5], the model of desire-belief-intentions of Rao and Gorgeff [10] and the temporal logic of beliefs of Michael Wooldridge [16]. Unfortunately, specify formally systems using temporal logic can become a difficult work, mainly because the notation is different to that used in programming languages. The objective of LTLAS is to keep the power of temporal logic but offering an easy interpretation to Agent-base systems specifications. Because LTLAS is equivalent to AML, it is possible to say that LTLAS translate, the AML which is difficult to express and understand, in order to give it a syntactical structure and an associated semantic easy to understand.

The organization of this article is: Section 2 describes AML Meta- Language; section 3 presents LTLAS language and section 4 is dedicated to show an application and finally in section 5 we give our conclusions.

2. AML Formal Model

In order to remember the way in which the logic can be used to describe dynamic properties of agents here we remember briefly the Agent Logic or AL. The AL represents the base of the Agent Meta Language AML giving a formal model to study the characteristics, properties and behavior of Multi-Agent Systems, before implementing it as a computational system. The temporal logic used for AML is a kind of non-classical logic [16] where the time model offers the basis to describe dynamical systems. In this paper are used time linear discrete sequences as a basic temporal model. Using this temporal model is possible define an agent as the 4-tuple:

$$A \overset{\text{def}}{=} < \beta^o, \mathcal{A}, M, \eta>$$

Where:

$\beta^o \in BS$ is the agent's initial belief set.

$\mathcal{A}$: $BS \rightarrow Ac$ is the agent's action set.

M : $BS \rightarrow \wp(Mess)$ is the agent's message generation function.

η : $BS \times Ac \times \wp(Mess) \rightarrow BS$ is the agent's next state function.

Using the previous agent's definition it's possible to specify agent's characteristics and observe its behavior and state changes through actions executed by the agent. Taking agent's initial belief an agent selects an action to execute using the function $\mathcal{A}$ and some messages to send using the function M. The function η then change the agent from one state to another, on the foundation of received message and the action it has just achieved.

On basis of previous agent's definition, we give the Multi-Agent definition as an indexed set of such agents:

$$MAS \overset{\text{def}}{=} \{ < \beta^o_i, \mathcal{A}_i, M_i, \eta_i > : i \in \mathbf{Ag} \}$$

The linear time temporal belief logic is a branch of modal logic; it is associated with time situations development that is, dynamic situations where the formulae interpretation is accomplished using frames with linear accessibility relations.

The AL [16] is a prepositional logic and contains three atomic operators: *Bel* to describe agent's Belief, *Send* to describe the messages that agents send and *do* for describing the actions that agents execute. Likewise, AL contains a set of modal temporal operators, which allows the description of agent's dynamic properties

2.1 Syntactic rules for AL

AL allows understand the agent's Belief, actions and the messages they send. To express the agent's beliefs are defined the internal language L in which AL is based. The language *AL* based on *L* contains the next symbols:

- The symbols { True, Bel, Send, Do }
- A countable set of constant symbols Const made up of disjoint sets $Const_{Ag}$ (agent constant) y $Const_{Ac}$ (action constant).
- All the closed formulae of internal language L
- The unary propositional connective $\neg$ and the binary connector $\vee$
- The unary temporal connectives $\{ O, \otimes \}$ and the binary temporal connectives $\{\mu, \omega\}$
- The punctuation symbols: {,), (and }.

The set of well-formed (wff) formulae of AL based on internal language L are defined by the next rules:

1. - if *i, j* are agent **ids**, ϕ is a closed formulae of L and α in an action constant then the next are atomic formula of L:

$$\text{true} \qquad (\text{Bel } i, \phi) \qquad \text{Send}(i, j, \phi) \qquad (\text{Do } i, \alpha)$$

2. - if ϕ and ψ are formulae of AL, then the following are formulae of AL:

$$\neg\phi \qquad \phi\vee\psi$$

3. - if ϕ and ψ are formulas of AL, then the following are formulae of AL:

$$O\phi \qquad \otimes\phi \qquad \phi\,\mu\,\psi \qquad \phi\,\omega\psi$$

The first four rules deals with atomic formulae of Al, the formulae true is a logical constant for truth, its always satisfied. The formula (Bel i, ϕ) is read: agent i belief ϕ. The formula (Do i, α) is read: agent i perform the action α. The formula Send(i, j, ϕ) describe the sending of messages and will be satisfied if agent *i* has sent *j* a message with content ϕ.

2.2 Semantic rules for AL

The satisfiability relationship ($\models$) for the LA, is given among evens in the way $< M, u>$ and formulas of LA. Given a model M and a temporary formula ϕ, then we say that ϕ is satisfied in a time position $u \geq 0$ of M and we denote it by: $< M, u> \models \phi$ [8].

In order to give a representation and interpretation to AL expressions, also to verify that a Multi-Agent specification is accurate, we use the semantic rules displayed in Fig. 1.

$< M, u> \models \text{true}$

$<M, u> \models (\text{Bel } i, \phi)$ iff $\phi \in \text{bel}(\sigma, I(i), u)$

$< M, u> \models (\text{Do } i, \alpha)$ iff $\text{action}(\sigma, I(i), u) = I(\alpha)$

$< M, u> \models (\text{Send } i, j, \phi)$ iff $< I(i), I(j), \phi> \in \text{sent}(\sigma, u)$

$< M, u> \models \neg\phi$ iff $< M, u> \not\models \phi$

$< M, u> \models \phi \vee \psi$ iff $< M, u> \models \phi$ or $< M, u> \models \psi$

$< M, u> \models O\phi$ iff $< M, u+1> \models \phi$

$< M, u> \models \otimes\phi$ iff $u>0$ and $< M, u-1> \models \phi$

$< M, u> \models \Box\phi$ iff $\forall\, k \geq u \quad < M, k> \models \phi$

$< M, u> \models \Diamond\phi$ iff $\exists$ un $k \geq u$ s.t $< M, k> \models \phi$

$< M, u> \models \phi\,\mu\psi$ iff $\exists\, k \in N$ s.t. $k \geq u$ and $< M, k> \models \psi$ and
$\qquad \forall\, w \in \mathbb{N}$ s.t $u \leq w < k$ and $< M, u> \models \phi$

$< M, u> \models \phi\omega\psi$ iff $< M, u> \models \phi\,\mu\psi$ or $< M, u> \models \Box\phi$

$< M, u> \models \upsilon\phi$ iff $\forall\, 0 \leq k \leq u \quad < M, k> \models \phi$

$< M, u> \models \blacklozenge\phi$ iff $\exists\, 0 \leq k \leq u \quad < M, k> \models \phi$

$< M, u> \models \phi \Rightarrow \psi$ iff $< M, u> \models \neg\phi \vee \psi$

Fig. 1 Semantic rules for AL

3. LTLAS Description

LTLAS is language equivalent to AML, it is useful to specify formally multiagent systems. The main advantage of LTLAS regarding AML is the possibility to specify multiagent systems easily because; LTLAS looks like a common programming language. The sentences of LTLAS are derived from the semantic rules for AL, discussed above in 2.2. In this way both languages are equivalent. LTLAS is visualized as a set of sentences. The propose of sentence actions are first: execute an agent action, e.g. (Do i, α); to send a message e.g. Send(i, j, ϕ) or to storage a belief e.g. (Bel i, ϕ). Second give a syntactic structures (conditions an cycles) to LTLAS e.g. While-Do (ϕ μ ψ).

The action sentence (Do i, α) involve primitive actions, also called constant actions, in this case α is a primitive action. The primitive actions belong to a set of actions given by KQML [14] and LCIASA[3] messages sets.

3.1 LTLAS Syntax

The syntax is considered as the group of rules that generate the correct sentences of the language, including particulars like the placement of words and the punctuation signs. In this paper we use an axiomatic notation to describe the language. Our language LTLAS provides the syntactic structure and the semantic interpretation to the actions expressed in the AML. That is, with LTLAS a computational interpretation is given to the formal agents model proposed in AML and it corresponds to the Object Language.

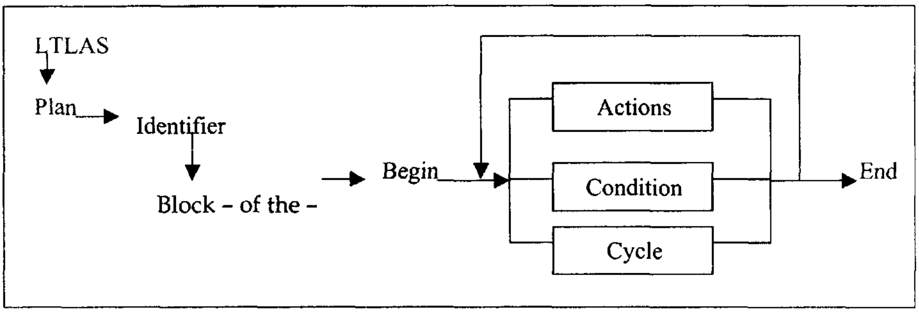


Fig. 2 General Structure of the LTLAS syntax

The interaction mechanism among the agents is carried out by means of the exchange of messages, which are grouped to form a plan, this plan is perceived as a sequence of messages to realize a certain task. The syntax of LTLAS is based on the plan concept. The Fig. 2 shows in a general way, the syntax of LTLAS, which is based on the message concepts and of plans. Each plan has an identifier that characterizes it, a block-of the-Plan that contains the attributes of the plan, its pre-conditions, the body of the plan and its post-conditions.

In this paper we use an axiomatic approach to define the syntactic structure of LTLAS. This approach is based in the following general notation for an axiom:

$$a; b; c; .... g \ \ |\text{-} \ z$$

Which is reading as: from the premises a, b, c, ...g can be asserted z. Moreover in the derivation process we use the inference rule called modus ponens ($|\text{-}\ \phi\ \ |\text{-}\phi \Rightarrow \psi\ \ |\ \ |\text{-}\psi$). In this approach we use the semantic rules given above in 2.2 for the AL language as premises of the axiomatic system in order to derive a set of rules as an axiomatic system, which guarantee the specifications correctness.

The Fig. 3 displays the syntactic structure and the semantic interpretation of LTLAS, using an axiomatic approach to describe it.

$\langle M, u\rangle = T \vdash \langle M, u\rangle \models \text{true}$	True-value
$\phi \in \text{bel}(\sigma, I(i), u) \vdash \langle M, u\rangle \models (\text{Bel } i, \phi)$	Bel-action
$\text{action}(\sigma, I(i), u) = I(\alpha) \vdash \langle M, u\rangle \models (\text{Do } i, \alpha)$	Do-action
$\langle I(i), I(j), \phi\rangle \in \text{sent}(\sigma, u) \vdash \langle M, u\rangle \models (\text{Send } i, j, \phi)$	Send-action
$\langle M, u\rangle \not\models \phi \vdash \langle M, u\rangle \models \neg\phi$	Neg-action
$\langle M, u\rangle \models \phi$ or $\langle M, u\rangle \models \psi \vdash \langle M, u\rangle \models \phi \vee \psi$	or- operator
$\langle M, u+1\rangle \models \phi \vdash \langle M, u\rangle \models O\phi$	Next-ϕ
iff $u>0$ and $\langle M, u-1\rangle \models \phi \vdash \langle M, u\rangle \models \otimes\phi$	Prev-ϕ
$\exists k \in N$ s.t. $k \geq u$ and $\langle M, k\rangle \models \psi$ and $\forall w \in N$ s.t $u \leq w < k$, $\langle M, w\rangle \models \phi \vdash \langle M, u\rangle \models \phi \, \mu \, \psi$	While- Do
$\langle M, u\rangle \models \phi \mu \psi$ or $\langle M, u\rangle \models \Box\phi \vdash \langle M, u\rangle \models \phi \omega \psi$	Do-cycle
$\langle M, u\rangle \models \neg\phi \vee \psi \vdash \langle M, u\rangle \models \phi \Rightarrow \psi$	If-action
$\vdash a+b+c...z+A+B...+Y+Z$	letter
$\vdash 0+1+2..9$	digit
$\vdash ++-+/+*$	Aritm-op
$\vdash < + = + > + \neq + \geq + \leq$	Rel-op
$\vdash$.	Point
$\vdash$,	Sc.

r letter $\vdash$ r identifier

s letter; t identifier $\vdash$ st identifier

x digit $\vdash$ x number

x digit, y number $\vdash$ xy number

x idenfitier; y Action $\vdash$ xy Agent-Action

r Agent-Action; m Sc; s Bool-Exp $\vdash$ Bel(rms) Agent-Belief

j Bool-exp $\vdash$ if j IF-exp

x Bool- exp $\vdash$ then x Then-exp

r IF-exp; s Then-exp $\vdash$ rs If-Then-exp

m identifier $\vdash$ Plan m heading-of-the-plan

x pre-condition; y declaration-of-variables; w body-of-the plan, z post-conditions $\vdash$ Begin xywz Block-of the-plan End

x heading-of-the-plan; y block-of-the-plan $\vdash$ xy Plan

Fig. 3 LTLAS syntax and semantic description

4. Application: Electronic Commerce

In order to illustrate the use of LTLAS, we present an application we implement in Java to test our language for specification. This example is about the electronic commerce application displayed in Fig. 4, which represent a dialogue being held by agents. The purpose of this agent's dialogue is to acquire a product through a messages interchange and obtain in this way a concordant solution. The interaction process presented by a Dooley graphic [1,2], and modified by Parunak. In order to simplify the actions sequence we consider the principal actions only. In the graphic representation are listed two kinds of agents: an client agent (Purchaser) denoted by A, and three suppliers agents denoted by B, C, D (Sellers), A_i, B_i, C_i, D_i denote subsequent states of A, B, C and D respectively.

In our application we identify the next four stages: *First;* find suppliers, steps 1-9 in Fig. 4, In this phase the client explores the products' description of a list of suppliers and in function of the parameters like the price, cost, quality of the product, time of delivery and quality of service, selects to the provider most appropriate.

Second; term agreement, steps 10-11 in Fig. 4, In this step the buyer revises the salesperson's catalogue and he makes an order, he generates and fills the pre-purchase form, the salesperson calculates the taxes, shipping and handling and he sends the OBI petition to the buyer, the buyer approves the order, he carries out the payment and he sends the complete order to the salesperson.

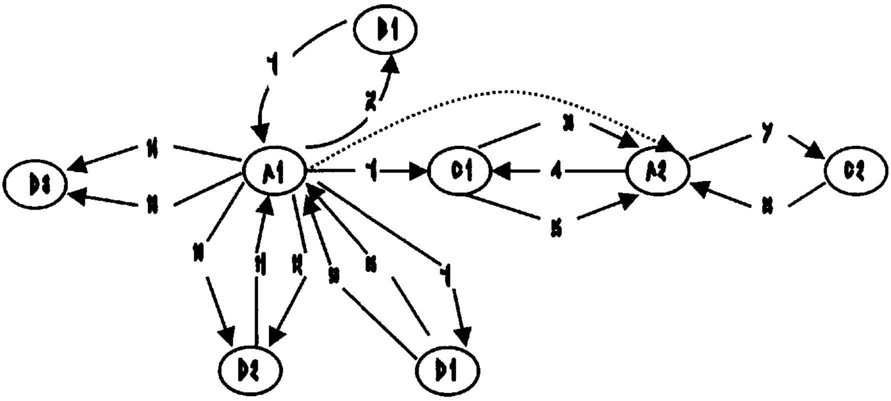


Fig.4 Product acquisition diagram

Third; supply the order, step 12 in Fig. 4 In this stage the salesperson requests the credit authorization to the bank, he registers and notifies the order, if he has the authorization of the credit and the product is available the salesperson orders the product shipment to the client.

Fourth support the client, steps 13, 14 in Fig 4. In this stage the order and the client personal data are registered to take it like reference in future operations.

Next, in fig. 5 we expose the Find Suppliers phase, using, LTLAS to derive the actions sequence associated with these phases of the application.

```
Plan Find-suppliers
 Begin
   Variables Catalogue, P, S1, S2, S3, F : Agents
   Preconditions: KQML protocol
    Begin-Plan
      While (( Catalogue = True) ∧ ¬Eos )
         Send (P,F, recruit-all (ask-all(x))) ∧ (Bel F Authenticates-Originator (P))
```

```
        Do (P, recruit(ask-all(x)))
        Send (S1,F, Advertise (ask (x))) ∧ (Bel S1 Authenticates-Originator (S1))
        Do (S1, Advertise(ask (x))
        Send (S2,F, (Advertise (ask (x))) ∧ (Bel S1 Authenticates-Originator (S2))
        Do (S2, Advertise(ask (x)))
        Send (S3,F, Advertise (ask (x))) ∧ (Bel S1 Authenticates-Originator (S3))
        Do (S3, Advertise(ask (x))
        Send (F,S,  broadcast(ask (x))) ∧ (Bel S2 Authenticates-Originator (F))
        Do (F, ask(x))
        Send (S1,P, Tell(x)) ∧ (Bel P Authenticates-Originator (S1))
        Do (S1, Tell(P))
        Send (S2,P, Tell(x)) ∧ (Bel P Authenticates-Originator (S2))
        Do (S2, Tell(P))
        Send (S3,P, Tell(x)) ∧ (Bel P Authenticates-Originator (S3))
        Do (S3, Tell(P))
        If (Val(S1) >((Val(S2) and Val(S2))) then
        Send (P, S1,  Tell(x)) ∧ ((Bel S1 Authenticates-Originator (P))
        Do (P, Tell(x)
    Do (Cheks-Plan)
  End -Plan
End
```

Fig.5 Find suppliers Specification phase in LTLAS

In this description, we use the properties of liveness and safety [11,4] to assure the consistency of the sequence of actions corresponding to this phase of the CB protocol. The specification in LTLAS looks like any programming language we know today. However, it keeps all the power of specification of AML. In this way LTLAS can be seen as a translation of AML.

5. Conclusion

In this article we present LTLAS a language useful to facilitate the specification of agent-based systems. As showed in the previous sections, LTLAS application result suitable to specify Multi-Agent system, because LTLAS sentences offer a Multi-Agent specification expressive, trusted, legible and easy to use and understand. LTLAS is suitable to develop cooperative applications as: selling-buying operations, bank transactions, electronic commerce, and industrial applications.

The associations of LTLAS with AML offer another perspective for Multi-Agent specification. Because is possible check the application consistence, the system behavior and the characteristics of a specification for a Multi-Agent system applications, before implementing it as a computational system.

Our future work is directed mainly towards: verification task process, in order to assure the specification correctness. We are studying the possibility for using the STEP demonstrator described in [13] or the Rao & Georgeff Algorithm [9]. Also we start developing a tool for automatic programming, that is, implement a tool for translating the specification to an implementation in Java.

6. References

[1] Barthes, J.P. *Multi-Agent systems, International Symposium on Advanced Distributed Systems*, Guadalajara, Jal. Marzo de 2000.
[2] Dooley, R.A. *"Appendix B-Repartee as a graph", an anatomy of speech Notions*, pp 384-386, 1976.
[3] Farías M.N. LCIASA: *Lenguaje de " Capacidad de Interacción " de Agentes en Sistemas Abiertos*, M. Sc. Thesis degree, CINVESTAV-IPN Guadalajara unit, México, 2000.
[4] Galton A. Temporal Logics and their applications, Academic Press, 1987
[5] G.M.P. O' Hare, and N.R. Jennings, *Foundations of distributed Intelligence*, Jhon Wiley & Sons Inc.
[6] Haddadi A., Sundermeyer K. " Chapter 5 –Belif-Desire-Intentions Agent Architectures, " *Foundations Of Distributed Artificial Intelligence.*, G.M.P. O'Hare and N.R. Jennigs (eds) Jhon Wiley & SonsInc., pp 169-186 1996.
[7] Jennigs, N.R., " Specification and Implementation of a Belief Desire Joint Intention Architecture for Colaborative Problem Solving," *Journal of Intelligent and Cooperative Information Systems* 2 (3), pp. 289-318, 1993
[8] Michael Wooldidge, Jorg P. Muller, Milind Tambe, *Intelligent Agents, Vol I,II, III*, Springer Verlang.
[9] Rao A.S. and Georgeff, " A Model-theoretical approach to the verification of situated reasoning systems", Proc. of The 13 Int. Joint on artificial intelligence, 318-324 Chambery, France.
[10] Rao, A.S. & Georgeff, M.P., "BDI Agents: From Russell Norvig, *Artificial Intelligence: a Modern Approach*, Prentice Hall., 1996.
[11] Sape Mullender, *Distributed Systems.*, Addison Wesley, 1993.
[12] Shoham Yoav, Agent Oriented Programming. In [45], pages 329-349. 1997 (reprinted from Artificial intelligence, 1993).
[13] STEP Automatic Theorem Demonstrator, developed by the Stanford University USA, available from http://rodin.stanford.edu
[14] Tim Finn, Don Mc Kay, Rich Fritzon, *AN OVERVIEW OF KQML: A KNOWLWDGE QUERY AND MANIPULATION LANGUAJE*, University of Maryland, Baltimore MD 21228
[15] Thomas R., The PLACA agent programming language. In intelligent agents: Agents theories, architectures and languages, pages 335-370. 1995.
[16] Wooldridge, M., " Chapter 10 –Temporal Belief logic for modeling Distributed Artificial Intelligence systems, " *Foundations Of Distributed Artificial Intelligence.* G.M.P. O'Hare and N.R. Jennigs (eds) Jhon Wiley & SonsInc., pp 269-285 1996.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Fuzzy Identification of a pH Neutralization Process

Rosimeire A. Jerônimo, Ilana A. Souza, Edson P. Maldonado
Centro Universitário São Camilo
Ciência da Computação
Av. Nazaré, 1501 - Ipiranga - 04263-200
São Paulo - SP - Brazil
E-mail: rosi_jeronimo@yahoo.com.br, iasouza@terra.com.br, puig@uol.com.br

Abstract: This article investigates the application of TSK-type fuzzy models to identify a simulated pH neutralization process. The pH neutralization process is considered a very important problem in the process industry, due to its many applications. The pH modeling is presented through the reaction invariants method. A "*Modified Gram-Schmidt*" (*MGS*) orthogonal estimator is used to estimate the consequent parameters. The obtained results of the models fuzzy are analyzed and then cross-validated.

Keywords: Fuzzy models, structure detection, parameter estimation, System identification.

1 Introduction

System identification aims to obtain models that reproduce the dynamic behavior of a process based on input and output data. The resulting models may be linear or nonlinear. The linear models are normally easier to obtain, since the linear identification techniques are well developed and the model is simpler. On the other hand, it is more difficult to get nonlinear models, due to the variety of alternatives of structures and the complexity of selecting the relevant terms. Identification of nonlinear is a relatively new area of research. The theoretical basis of nonlinear identification is still being established.

In most studies of identification of a process by using its input-output data, it is assumed that there exists a global functional structure between the input and the output such as a linear relation.

It is, however, very difficult to find a global functional structure for a nonlinear process. On the other hand, fuzzy modeling is based on the idea to find a set of local input-output relations describing a process. So it is expected that the method of fuzzy modeling can express a nonlinear process better than an ordinary method.

The system identified in this work is the model of a pH neutralization process occurring inside a CSTR (continuous stirred tank reactor) which has many industrial applications.

This article is organized as follows: in the 2nd section a abbreviation description of the pH neutralization process is presented. The description of TSK-type models for dynamic systems is introduced in the 3rd section. The identification of the process of pH neutralization is accomplished and the obtained models are presented in the 4th section. Finally, the 5th section presents the conclusions.

2 Description of the pH Neutralization Process

The pH neutralization process is a non-linear process and it has a high gain next to the neutrality point (pH=7), where the slope of the titration curve is very high. Another characteristic of this process is that the composition of the incoming fluid is normally variable. The process dead time is variable and it depends on the volume variation and on the process input flows [2]. Furthermore, small quantities of intruder elements such as, for example, carbon or potassium (with buffering capacity) change the process dynamics.

The pH neutralization plant used to perform the work is a CSTR unit, in which it is desired to obtain an effluent with a desired pH (normally neutral) by introduction of a base in a generic acid. A simplified diagram of this plant is shown in figure 1, which is an extension of the process studied by [3], [1] and [7].

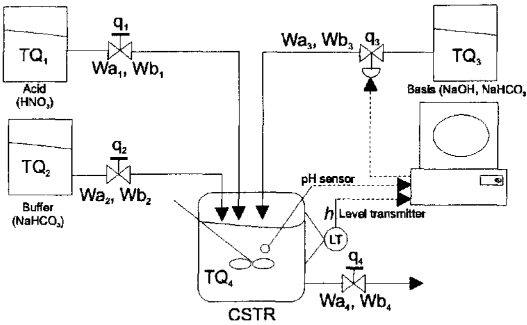


Figure 1: Simplified diagram of a pH neutralization process

In the process, there are three input flows (q_1, q_2 and q_3) which are mixed in tank TQ_4 and an output flow q_4, representing the flow of the output fluid. In this work, two measured inputs and one output variable were considered for the process identification. The variables of considered inputs considered in the identification process were: q_1 and q_3. The output variable was the pH.

The variables of the pH neutralization process and their values nominal are shown below. Inputs:

- q_1 = acid flow (nitric acid, HNO_3), ($\bar{q}_1 = 0,9996$ l/min);
- q_2 = buffer flow (sodium bicarbonate, $NaHCO_3$), ($\bar{q}_2 = 0,333$ l/min);
- q_3 = base flow (sodium hydroxide mixture, $NaOH$ with sodium bicarbonate, $NaHCO_3$), ($\bar{q}_3 = 0,4686$ l/min);
- C_1 = acid concentration; ($\overline{C}_1 = 2$ M);
- h = liquid level inside the CSTR, ($\bar{h} = 0,294$ m).

Output:

- pH = pH solution inside the CSTR, ($\overline{pH} = 7$).

The equations that describe the process as well as their parameters and the chemical reactions are here omitted but can be found in [6].

3 Fuzzy Modeling

Takagi-Sugeno fuzzy models are suitable to model a large class of nonlinear systems. Fuzzy modeling from measured data are effective tools for the apprroximation of uncertain nonlinear systems.

To build fuzzy models from data generated by poorly understood dynamic systems, the input-output representation is often applied. A very common structure is the NARX model ("Non-Linear Auto-Regressive with eXogenous inputs"), which can represent the observable and controllable models of a large class of discrete-time non-linear systems. It establishes a relationship between the collection of past input-output data and the predicted output.

$$\begin{aligned}\hat{y}(k+1) =& F(y(k), \dots, y(k-n_y+1), u(k), \dots, \\ & u(k-n_u+1))\end{aligned} \tag{1}$$

where k denotes discrete time samples, n_y and n_u are integers related to the system order.

Consider the nonlinear multi-input single-output system

$$y = f(x_1, x_2, \dots, x_n) \tag{2}$$

which has known m operating points $(x_{i,1}, x_{i,2}, \dots, x_{i,n})$, $i = 1, \dots, m$, around which the system can be decomposed into m linearized subsystems: $y_i = f_i(x_1, x_2, \dots, x_n)$. The TSK model for the system can be generically represented as

$$\begin{aligned} R_i: \ & \text{if} \quad x_1 \quad \text{is} \quad A_{i,1} \quad \text{and} \dots \text{and} \quad x_n \quad \text{is} \quad A_{i,n} \\ & \text{then} \quad y_i = f_i(x_1, x_2, \dots, x_n) \end{aligned} \tag{3}$$

or, more specifically:

$$\begin{aligned} R_i: \ & \textbf{if} \quad y(k) \quad \textbf{is} \quad A_{i,1} \quad \textbf{and} \quad y(k-1) \quad \textbf{is} \\ & A_{i,2} \quad \textbf{and} \dots \quad \textbf{and} \quad y(k-n_y+1) \\ & \textbf{is} \quad A_{i,n_y} \quad \textbf{and} \quad u(k) \quad \textbf{is} \quad B_{i,1} \quad \textbf{and} \\ & u(k-1) \quad \textbf{is} \quad B_{i,2} \quad \textbf{and} \dots \textbf{and} \\ & u(k-n_u+1) \quad \textbf{is} \quad B_{i,n_u} \quad \textbf{then} \\ & \hat{y}_i(k+1) = \sum_{j=1}^{n_y} a_{i,j} y(k-j+1) + \\ & \sum_{j=1}^{n_u} b_{i,j} u(k-j+1) + c_i \end{aligned} \tag{4}$$

where $a_{i,j}$, $b_{i,j}$ and c_i are the consequent parameters. The NARX local model can represent MISO systems directly and MIMO systems in a decomposed form as a set of coupled MISO models. It should be noted that the dimension of the regression problem in input-output modeling is often larger than the state-space models, since the state of the system can usually be represented by a vector of a lower dimension than, for instance, in the NARX model given by equation (1).

4 pH Neutralization Process Identification

The pH neutralization process identification was performed in open loop. The consequent part of parameter identification was made *"off-line"*, combining the *"Modified Gram-Schmidt"* method (its description can be found in [6]) with the method presented in [5]). The model significant terms can be selected based on the *"error reduction ratio"* (ERR).

4.1 Project of the Identification Experiment

The data were acquired during 1000 minutes with a sampling time $T_{ss} = 20/60$ minutes ($\cong 0,333$ min), which corresponded to 2001 samples. The inputs were disturbed with evenly distributed random signals and they occurred simultaneously in all the input variables. The measured inputs are: acid flow (q_1) and base flow (q_3). The pH value (pH) was the output variable. The other variables of the system, i.e., were kept constant at their nominal values.

2001 samples different from the identification data were used to do the validation of the model.

4.2 *Non-Linearity Detection*

It is very important to check the existence of nonlinear interactions in the identification data. There are algorithms that detect the non-linearities and indicate if it is necessary to use nonlinear models to reproduce the system dynamics.

Consider the dynamic system described in the following dynamic equation:

$$\begin{aligned} y(t) =& f^l[y(t-1), y(t-2), \dots, y(t-n_y), u(t-d), u(t-d-1), u(t-d-2), \dots, \\ & u(t-d-n_u+1), e(t-1), e(t-2), \dots, e(t-n_e)] + e(t) \end{aligned} \tag{5}$$

where f is a generic mathematical function. The sampling period T_s was normalized. [4] demonstrated that:

$$\phi_{y'y^{2'}}(\tau) = E\left\{\left[y(t) - \overline{y(t)}\right]\left[y^2(t-\tau_c) - \overline{y^2(t)}\right]\right\} = 0, \qquad \nabla\tau_c \tag{6}$$

Equation (6) is valid if and only if the system given in equation (5) is linear. In this case, the system can be considered linear if equation (6) is kept within the probabilistic confidence interval of 95%, that delimits the region where the autocorrelation function given above has to remain to be considered null. The limits of the confidence interval are approximately $\pm 1,96/\sqrt{N}$, being N the length of the available data vector.

The 2001 output data collected for the identification process were used to detect if there were nonlinear interactions in the identification data. Figure 2 illustrates the obtained result.

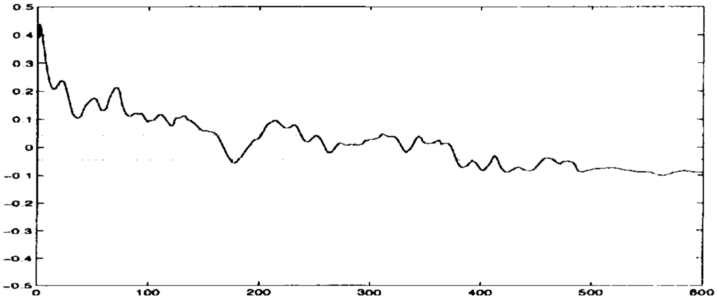


Figure 2: Non-linearity detection test (pH)

The result shown in figure 2 presented expressive non-linearities out of the confidence interval of 95%, indicating that nonlinear structures are necessary to represent the pH neutralization process dynamics.

4.3 *Fuzzy Identification*

For the of input variables, acid flow (q_1), the fuzzy linguistic attributes were: very low (MB)0, low (B), medium (M), high (A), very high (MA). The acid flow varies between $(0-2)\ l/min$, whereas the base flow varies between $(0-1.7)\ l/min$. The output variable, pH, varies between $(0-14)$ with the following attributes linguistic: pH very acid (pH MA), pH little acid (pH A), pH medium (pH M), pH little basic (pH PB), pH very basic (pH MB).

Figure 3illustrates the membership functions of the input variables for the acid flow (q_1) and base flow (q_3), respectively.

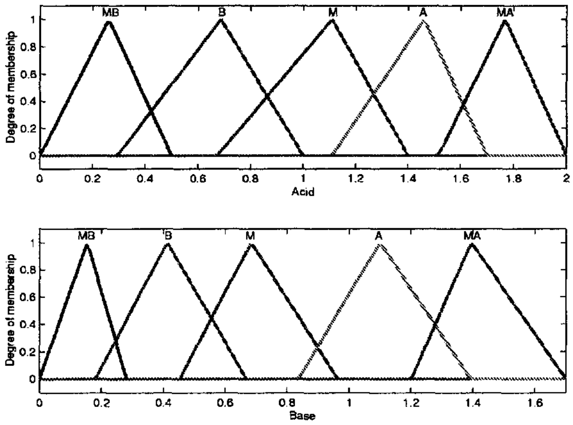


Figure 3: Membership functions of the input variables, (acid flow) and (base flow)

Figure 4 illustrates the membership function of the output variable for the pH.

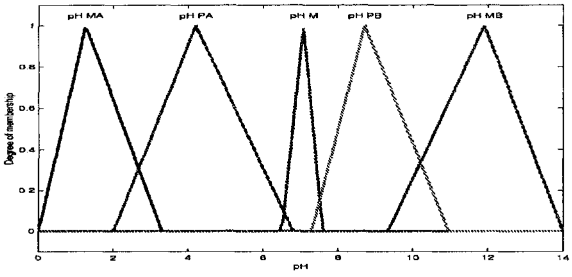


Figure 4: Membership function of the output variable, (pH)

The process is represented as a first order discrete-time NARX model

$$pH(k+1) = f(pH(k), q_1(k), q_3(k)) \tag{7}$$

where k denotes the sampling instant, f is an unknown relationship approximated by the TSK fuzzy model, $q_1(k)$ and $q_3(k)$ represents the acid flow and base flow, respectively and $pH(t)$ represents the output variable, pH.

The figure 5 illustrates the graphic of surface of the process of pH neutralization.

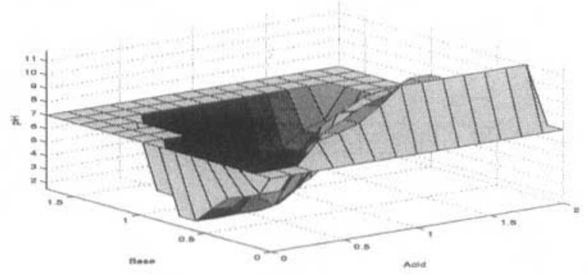


Figure 5: Visualization of the surface of the process of pH neutralization

The model significant terms were selected based on the "*error reduction rate*", resulting in only 5 rules to describe the identification procedure considered in this work. The resulting rules were:

R_1 : if $pH(k)$ is MA and $q_1(k)$ is MB and $q_3(k)$ is M
then $pH(k+1) = 0.013 + 0.89930pH(k) + 0.7890q_1(k) - 0.5648q_3(k)$
R_2 : if $pH(k)$ is M and $q_1(k)$ is Mand $q_3(k)$ is B
then $pH(k+1) = 0.467 - 0.6579pH(k) + 0.2310q_1(k) + 0.13468q_3(k)$
R_3 : if $pH(k)$ is PB and $q_1(k)$ is A and $q_3(k)$ is B
then $pH(k+1) = 0.1234 + 2.0124pH(k) - 0.4508q_1(k) + 0.2352q_3(k)$
R_4 : if $pH(k)$ is MB and $q_1(k)$ is MA and $q_3(k)$ is MB
then $pH(k+1) = 1.2345 + 1.0867pH(k) + 0.1569q_1(k) + 0.0985q_3(k)$
R_5 : if $pH(k)$ is M and $q_1(k)$ is PA and $q_3(k)$ is PB
then $pH(k+1) = 2.9781 + 1.0982pH(k) + 0.3489q_1(k) + 0.3467q_3(k)$

The figure 6 show the cross-validation graphic of the fuzzy model.

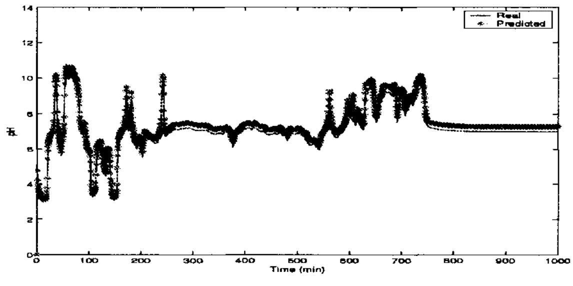


Figure 6: Model validation of the process of pH neutralization

5 Conclusions

Nonlinear fuzzy models were here employed to identify a pH neutralization process. The application of an advance regression orthogonal estimator, based on the orthogonalization procedure named Modified Gram-Schmidt (*MGS*), resulted in nonlinear fuzzy models that satisfactorily represented the pH neutralization process dynamics.

References

[1] D. M. Girardot, Control of pH based on Reaction Invariants, Master Dissertation, University of California, Santa Barbara, (1989).

[2] O. A. Zanabria, Modeling, Simulation and Control of pH Neutralization Process, Master Dissertation, Polytechnic School of the University of São Paulo, Department of Eletronic Engineering, Laboratory of Automation and Control, São Paulo, SP, Brazil, (1997).

[3] R. C. Hall, Control of pH Based on Reaction Invariants, Master Thesis, University of California, Santa Barbara, (1987).

[4] S. A. Billings and S. F. Voon, Structure Detection and Model Vality Tests in the Identification of Nonlinear Systems, IEEE Proceedings **130** Parte D, (1983) 193–199.

[5] L. Wang and R. Langari, Building Sugeno-Type Models Using Fuzzy Discretization and Orthogonal Parameter Estimation Techiques, IEEE Transcations on Fuzzy Systems **3** (1995) 454–458.

[6] R. A. Jerônimo, Modeling, Non-linear Identification of a Multivariable pH Neutralization Process Using Polynomial Models, Master Dissertation, Polytechnic School of the University of São Paulo, Department of Eletronic Engineering, Laboratory of Automation and Control, São Paulo, SP, Brazil, (1998).

[7] M. E. Henson, Feedback Linearization Strategies for Nonlinear Process Control, Master Thesis, University of California, Santa Barbara, (1992).

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

A Fuzzy Reed-Frost Model for Epidemic Spreading

Teresa E. P. Sacchetta[1,2], Neli Ortega[1*], Renée X. Menezes[1] and Eduardo Massad[1]
[1] *School of Medicine-LIM01/HCFMUSP, University of São Paulo*
[2] *Fleury - Centro de Medicina Diagnóstica*
Av. Dr. Arnaldo 455, São Paulo, 01246-903, SP, Brazil

Abstract

In this work, we propose a generalization of the classical Reed-Frost model considering heterogeneities in the population. In the Fuzzy Reed-Frost model the signals (symptoms, medical findings etc.) are included in the dynamics of the epidemic. Each individual presents a set of clinical indicators, which are used to determine his/her infectivity degree through a fuzzy decision process. The model is compared with a stochastic Reed-Frost model developed by some of the authors in a previous work.

Keywords: Fuzzy Relations, Reed Frost Model, Epidemiology, Beta distribution.

1. Introduction

Many models of epidemic spreading have been proposed to help in the comprehension of infectious diseases, with the obvious assumption that knowledge could help in the control of those diseases.

The Reed-Frost model was proposed by Lowell J. Reed e Wade H. Frost, after a series of lectures held at Johns Hopkins University, which was published later by their students Abbey[1] and Maia[2]. This model is a particular case of a chain binomial model, in which it is assumed that each infected individual infects susceptible individuals independently, and that individuals are under the same contact rate with each other. If we represent by p the probability of a contact between a susceptible and an infected individual resulting in a new case, we have that, at time t, the probability that a susceptible individual does become infected, C_t, is equal to the probability of at least one infectious contact, i.e.,

$$C_t = 1-(1-p)^{I_t}, t > 0,$$

where I_t is the number of infected individuals at time t. Time t is regarded as a discrete variable and an individual's classification, as either susceptible, infected or removed, changes only when time changes from t to $t+1$. Other assumptions are that the probability of an infectious contact is fixed along the epidemic course and is the same for all individuals.

The Reed-Frost model is useful for explaining epidemic spread in small groups and assumes that all the individuals have the same susceptibility and infectivity, what characterises the group as homogeneous. The effect of unapparent infections and

---

[*] email: neli@dim.fm.usp.br

underreporting cases is not considered in the model: no error is involved in the classification process, such as a truly infected individual being classified as susceptible.

Those assumptions of the classical Reed-Frost model do not apply in several real cases, for instance in diseases to which there is no diagnostic test readily or easily available, like influenza and other viral and bacterial infections. In those cases, the diagnostic process involves uncertainty and is based upon a set of clinical characteristics, often subjective, which we are going to call *signals*. The homogeneity is unlikely to hold in real epidemics; depending on the pathology, the symptoms vary during the epidemics, even in discrete-time ones, and the severity or type of the signal can interfere in the individual infectivity power. For example, an individual with cough or sneezing has greater chance of spreading the epidemic, while an individual that presents high fever may have his/hers infectivity decreased by being excluded of the contact with other individuals of the group.

In this paper, we propose a generalisation of the classical Reed-Frost model, including the clinical signals presented by each individual of the group in the classification process, through a fuzzy decision process. By doing this, we intend to incorporate individual heterogeneities in the classificatory process: signals are used to define whether an individual is infected or susceptible, and also to define how the epidemics will spread.

2. A Reed-Frost model with heterogeneity

Several attempts were made to generalise the Reed-Frost model, considering a non-homogeneous population in terms of infectivity or susceptibility [3-5]. But in all the attempts the solution was given by the stratification of the population into subgroups, considering homogeneous mixing within the subgroups. The groups are closed and the individuals remain in the same subgroup during the whole epidemic. In such situations, the level of infectivity and susceptibility is assumed as constant throughout the epidemic course, not with understanding heterogeneity among the different groups.

In a previous paper, Menezes et al. [6] included in the classical Reed-Frost model the uncertainty involved in the classificatory process, considering infectivity levels as a time-dependent random variable. Each individual presents a series of clinical indicators used to classify them as infected or not, and the classificatory process includes the possibility of errors, such as an infected individual being classified as susceptible. This model is a stochastic generalisation of a classical Reed Frost. Those authors [6] modelled an individual's infectivity as a function of the clinical signals, allowing for time-dependent, heterogeneous infectivity. Constant susceptibility levels are assumed and the probability structure lead them to an expression for the epidemic Basic Reproduction Ratio of the infection.

In this work we propose a generalisation of Menezes et al. [6] model, where the individuals classification process present a fuzzy decision structure based on clinical signals. We propose two different fuzzy structures and compare the simulation results with that of the Menezes et al. model.

3. Fuzzy Structure

Either in classical or Menezes et al. Reed-Frost models, the individuals should be classified in susceptible or infected. In the former this classification is performed by a random variable, which govern the dynamic of epidemic. In the latter this classification depends on the signals summary, which is a random variable itself. The generalisation consists in relaxing this crisp classification, proposing a membership degree, of each individual of the population, to the fuzzy set *Infected* and in the fuzzy set *Susceptible*, through a max-min composition of fuzzy relation.

Consider a set of signals **S** and the matricial representation of a fuzzy relation. Thus, **S** = $[s_{lk}]$ is the array of *k* signals of the individual *l*, **I** = $[i_{kl}]$ is the matrix that associate each signal to the Infective statement and **DI** = $[di_{ll}]$ is the membership degree of the individual *l* in the fuzzy set *Infected*. So, this fuzzy structure allows a fuzzy classification of each individual to an Infected class, providing a degree of infectiousness. In the same way, $\underline{\mathbf{S}}$ = $[\underline{s}_{lk}]$ is the array of *not signal*, i.e., $\underline{\mathbf{S}}$ is the negation of S, which could be computed through the *standard complement* [7] of the signals fuzzy set, S, using:

$$\underline{S} = 1 - S$$

for each signal. For instance, if the set of signal is **S** = [fever, cough], i.e., s_1 is fever and s_2 is cough, and an individual presents fever degree equal $s_1 = 0.7$ and cough degree equal $s_2 = 0.4$. So, the degree of *not fever*, $\underline{s}_1$, and *not cough*, $\underline{s}_2$, of this individual are $\underline{s}_1 = 0.3$ and $\underline{s}_2 = 0.6$, respectively. It is possible to find the degree of susceptibility to each individual computing the fuzzy relation between the fuzzy set *not signal* and the associative matrix *not signal/susceptibility* through max-min composition.

For example, consider the same two symptoms, *fever* and *cough*, and an array which represent the association of these symptoms and the infectiousness, $R_I = [r_{fever}, r_{cough}]$, where r_{fever} is the relationship degree between the symptom fever and Infected class and r_{cough} is the relationship degree between the symptom cough and Infected class. So, an individual that have a degree of fever, s_{fever}, and a degree of cough, s_{cough}, will belong to the infected fuzzy set with the degree given by:

$$DI = \max\{\min[s_{fever}, r_{fever}]; \min[s_{cough}, r_{cough}]\}.$$

The equivalent process is made by using each *not signal* degree and the relationship between the respective *not signal* and Susceptible class for calculating the degree of susceptibility,

$$DS = \max\{\min[\underline{s}_{fever}, \underline{r}_{fever}]; \min[\underline{s}_{cough}, \underline{r}_{cough}]\}.$$

We assume that each individual *i* has *k* signals, with levels represented by membership degree in each fuzzy subset of clinical signal (like fever, cough) s_{il}, s_{i2},...,s_{ik}. So, these levels are numbers between 0 and 1, with $s_{il} = 0$ indicating that the clinical signal 1 is absent in patient *i*, and $s_{il} = 1$ indicating that patient *i* presents the clinical signal 1 with maximum level (or severity). The classification of the individual *i* as infected or not is based upon the matrix that relates the signals and susceptible and infected status and the max-min fuzzy relation. The heterogeneity is considered in the epidemic dynamics through the signal influence on the parameters *P* (the probability of a infective contact between a susceptible and infected individual) and C_t (the risk of a susceptible individual becoming infected). The new individual set of signs in next time is found from C_t.

In order to compare this model with the structure proposed by Menezes et al., we present two kinds of generalization of both the Reed-Frost models. In all cases the membership degrees of each individual in the Infected and Susceptible fuzzy sets are computed by the max-min fuzzy relation described above, and the difference consists in the way this membership degree is used in the structure of the model.

3.1 Classification as Infected or Susceptible - model type 1

This model preserves the crisp classification of the individuals as Infected or Susceptible. However, the decision depends on which is the greater value among the membership degrees. The classification function, G_{it}, is given by:

$$G_{it} = \begin{cases} 1 & \text{if} \quad DI > DS \quad \Rightarrow \quad \text{Infected} \\ 0 & \text{if} \quad DI < DS \quad \Rightarrow \quad \text{Suscptible} \end{cases}$$

If $DI = DS$, then a homogeneous random number will decide about the individual classification. So, the number of infected individuals at time t is given by:

$$I = \sum_{i}^{N} G_{it}$$

At time t, the probability $P_{ij,t}$ that a contact between a susceptible individual i and an infected individual j results in an infective contact is computed by two forms. First, we assume the same probability function proposed by Menezes et al.. Thus, the only difference between their model and the present one is the classificatory method. In this case $P_{ij,t}$ is given by:

$$P_{ij,t} = \frac{\sum_{s}^{k} S_{I_s}}{zk},$$

where S_{Is} is the *sth* signal of the infected individual j, k is the number of signals and z is a parameter that makes $P_{ij,t}$ a small number, which is necessary for the operations. This $P_{ij,t}$ form is comparable to Menezes et al. model when $P_{ij,t} = D_{j,t}$, where $D_{j,t}$ is an average of the clinical signals.

In the second form of $P_{ij,t}$ we consider the infectiousness degree of the individual j, given by:

$$P_{ij,t} = \frac{DI_j}{z}$$

where DI is the infectiousness degree of the infected individual j.

In both cases the probability that a susceptible individual has, at time t, at least one infectious contact is:

$$C_t = 1 - \prod_{j=1}^{N} (1 - P_{j,t})^{G_{j,t}},$$

like in Menezes et al. stochastic Reed-Frost model.

3.2 Considering the individual degree of infectiousness - model type 2

In this case, we do not classify the individuals in infected or not, we just consider the degree of infectiousness of each individual. So the number of infected individuals, at time t, is found through the fuzzy cardinality of the Infected fuzzy subset, that is,

$$I = \sum_{l=1}^{N} DI_l$$

where N is a total number of individuals in the population (all individual is considered infected with some degree of infectiousness).

The probability $P_{ij,t}$ that a contact between a susceptible individual i and an infected individual j results in a infective contact of the form:

$$P_{ij,t} = \frac{DI_i + DI_j}{2z}$$

where $P_{ij,t} = P_{ji,t}$ and all possible contacts are considered. Since we are not using the information about the susceptibility degree of individuals, it is considered the homogeneity in the susceptible set.

Now, the probability of a susceptible individual having, at time t, at least one infectious contact is given by

$$C_t = 1 - \prod_{all\ pairs} (1 - P_{ij,t})$$

where there is no repetitive pair. Clearly this model is very different from Menezes et al. proposal.

3.3. Simulation

As expected, the preliminary results indicate that the behaviour of the epidemic depends on the studied region of parameters for all models. For certain regions in the space of parameters, Menezes et al. model provides extended epidemics, however this is not a rule, in other regions the epidemics are longer with the fuzzy model.

The model capacity of fitting for an epidemic clearly depends on the disease. Thinking of an influenza epidemic, in which the infectivity degree varies a lot depending on the symptoms, and considering that we are studying a small population, the fuzzy structure seems to be more appropriate, since a symptom summary is not a good indicator of the infective contact in this case. In order to simulate an influenza epidemic, inside a small group like a family for example, we consider four clinical signals: *fever*, *cough*, *sneeze* and *running nose*. The relational matrices *signal/infected*, *I*, and not *signal/susceptible*, *S*, were:

$$I = \begin{bmatrix} 0.8 \\ 0.6 \\ 0.7 \\ 0.6 \end{bmatrix} \qquad S = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.6 \\ 0.7 \end{bmatrix}$$

i.e., the relation between *fever* and to be infected is 0.8, *cough* and to be infected is 0.6, *sneeze* and to be infected is 0.7 and *running nose* and to be infected is 0.6. In the same way, the relation between *not fever* and to be susceptible is 0.8 (if the individual do not present fever, then the chance that he/she is susceptible to an infection is high), the *not* *cough* and to be susceptible is 0.7 and so on. To this case, in the fuzzy structure the epidemic spread quickly, resulting in a entirely susceptible population. The epidemic size depends on the severity of the clinical signals in the beginning of the epidemic course, as expected.

4. Discussion

Several attempts have been made to generalise the Reed-Frost model to consider a non-homogeneous group, either from the susceptibility or from the infectivity viewpoint [2,4,5,8]. In all these, the homogeneity assumption is relaxed by dividing the main group into subgroups, and considering homogeneous mixing within each subgroup. Subgroups are closed and individuals remain within the same subgroup for the entire duration of the epidemics, which means that an individual's susceptibility and infectivity levels are taken as constant throughout the epidemic course. To the best of our knowledge, Menezes et al. work [6] is the first Reed-Frost model that considers the heterogeneity of individuals as related to the clinical signals. The model proposed here provides generalisations forms to deal with those heterogeneities, particularly in the fuzzy classification method.

Menezes et al. model consider the arithmetic average of the symptoms as a signal summary to classify the individual as Infected or Susceptible. In Reed-Frost Fuzzy model type 1 this classification was performed based on the degree of infectiousness and susceptibility. In fact, the fuzzy structure do not require the classification of the individuals in infected or not in order to elaborate the dynamic of the epidemic. This consists in one of the main characteristic of fuzzy systems applied in epidemic studies. In the Reed-Frost Fuzzy model type 2, the epidemic dynamic is performed considering the membership infectiousness degree of all individuals. Apparently this approach is more useful for the majority of epidemiological diseases in reality, since in our model we aim to incorporate the individual heterogeneity according to which a person is infected or susceptible in different degrees, and considering the ability of each signal to spread the disease.

The fuzzy model structure presented allow also to compute the Basic Reproduction Ratio, R_0, defined by the number of secondary infections caused over one generation, after the introduction of a single infected individual in an entirely susceptible population [10]. In the type 1 model, R_0 would be calculated based upon the expected value of I_2 since I_1 is 1, and upon the probability distribution of the random variable *DI*, which is the infectivity degree. In type 2 it could be calculated through fuzzy clustering structures, as proposed in [11].

Finally, the development of new tools to deal with heterogeneities is getting more necessary. Including the several heterogeneities in our model we aim to incorporate naturally existing differences among individuals in order to make it applicable to real epidemic scenaria. For the one hand, there are several infections, like influenza, which, besides being transmitted among small groups of individuals, produce highly heterogeneous clinical pictures. On the other hand, the huge amount of genetic information provided by the emerging field of genomics (and proteonomics), generates clinical informations which may sharply distinguish individuals. We are convinced that the fuzzy structures would contribute with this new areas.

References

[1] ABBEY, H. An examination of the Reed-Frost theory of epidemics. *Human Biology* 24 (3):201-33, 1952.

[2] MAIA, J.O.C. Some mathematical developments on the epidemic theory formulated by Reed and Frost. *Human Biology* 24 (3):167-200, 1952.

[3] JACQUEZ, J.A. A note on chain-binomial models of epidemic spread: what is wrong with the Reed-Frost formulation. *Math. Bioscience* 87:73-82, 1987.

[5] LEFRÈVE C., PICARD P. On the formulation of discrete-time epidemic models. *Math. Bioscience* 95 (1):27-35, 1989.

[6] PICARD P., LEFRÈVE C. The dimension of Reed-Frost epidemic models with randomized susceptibility levels. *Math. Bioscience* 107 (2) : 225-33, 1991.

[7] MENEZES, R., ORTEGA, N.R.S. and MASSAD, E. A Reed-Frost Model Taking Account Uncertainties in the Diagnostic of the Infection. *Applied to Bulletin of Mathematical Biology*, 2002.

[8] KLIR, G. and YUAN, B.. Fuzzy Sets and Fuzzy Logic: Theory and applications. Prentice Hall, USA, 1995.

[9] SCALIA-TOMBA, G.. Asymptotic final size distribution of multityped Reed-Frost process. *Journal of Applied Probability* **23**: 563-584, 1986.

[10] DIEKMANN, O.; HEESTERBEEK, J. A. P. & METZ, J. A. J. (1990). On the definition and the computation of the Basic Reproduction Ratio R_0 in models for infectious diseases in heterogeneous populations. *Journal of Mathematical Biology* **28**: 365-382.

[11] MASSAD, E., ORTEGA, N., BURATTINI, M., STRUCHINER, C.. Fuzzy R_0 and Epidemic Dynamics. In: IMA - Conference on Mathematics Modelling and Statistical Analysis of Infectious Diseases, 2001, Cardiff – UK., 2001, p. 1-4.

Invited Talks

Abstracts

This page intentionally left blank

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Data Mining in Large Base using Rough Set Techniques

Germano Lambert Torres

Abstract:
The operation of a power system is intrinsically complex due to high degree of uncertainty and the large number of variables involved. The various supervision and control actions require the presence of an operator, who must be capable of efficiently responding to the most diverse requests, by handling various types of data and information .

One of the most important operator task is to determine the current operational state of the system. To accomplish this task, the operator receives many data measured into the system. These data can be analogical (e.g., position of circuit breakers) or digital (e.g., real power flow in a specific transmission line). By handling these data, the operator tries to built an image of the operation point of the power system supervised for him.

The analysis performed by the operator tries to make a classification of the operational state of the system in one of the three states: normal, abnormal, and restorative. In the first state, normal state, all loads are supplied and all measurements are inside of the nominal rates. In the second state, abnormal operational state, all loads continue to be applied but some of measurements are outside of the nominal rates. For the restorative operational state, some loads are not supplied, i.e., there was a loading shedding process.

The problem found by the operator is to use all available data in his analyses. A huge number of data and information in a control center database must be manipulated and , mainly, ,composed to allow the operator to have a visualization of the current state of the system. The manipulation of all data/information is not an easy task. This paper presents an example of an alternative approach to help the operators to produce the classification of the system in its three possible states. This approach is based on the Rough Set theory, proposed by Pawlak in 1982.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

An Automaton Model for Concurrent Processes

Manfred Droste

Abstract. Formal verification of concurrent reactive systems often leads to excessively large state spaces, the 'state space explosion problem'. Recently, several authors have proposed to exploit the possible independency of actions to consider only part of the global state space. Often, these independencies are dynamic, i.e. depend on the particular state of the system.

We will present an automaton model for such situations with varying independencies of actions. We will show how finite or infinite concurrent computations can be represented by labelled partial orders and how different logics (predicate calculus, temporal logics) can be used for specification of the partial orders and thereby of the computations. We develop connections to formal language theory, in particular for recognizable, aperiodic or star-free languages. This generalizes classical results of Kleene, B\"uchi, McNaughton and Papert, and recent results from trace theory. If time permits, we also point out a connection between our automaton model and Petri nets.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

It is a Fundamental Limitation to Base Probability Theory on Bivalent Logic

Lotfi A. Zadeh*

Extended Abstract

Probability theory has long played—and is continuing to play—the role of the principal tool for dealing with problems in which uncertainty is a significant factor. The history of probability theory is a history of important advances in our understanding of decision-making in uncertain environments. But what we see alongside the brilliant successes are problem areas in which progress has been elusive. An example is the class of problems in which probabilities, utilities and relations are ill-defined in ways that put such problems well beyond the reach of existing methods.

A thesis that is put forth in this paper is that standard probability theory, call it PT, has fundamental limitations—limitations which are rooted in the fact that PT is based on bivalent logic. It is this thesis that underlies the radically-sounding title of the paper: It is a Fundamental Limitation to Base Probability Theory on Bivalent Logic.

The principal rationale for this thesis is that the conceptual structure of PT is directed at addressing the partiality of certainty but not the partiality of possibility and, most importantly, the partiality of truth. A direct consequence of these omissions is that PT lacks the capability to deal with information which is not just partially certain but also partially possible and/or partially true. An example is: Most analysts believe that it is very unlikely that there will be a significant increase in the price of oil in the near future.

The principal negative consequences of basing PT on bivalent logic are the following.

1. Brittleness (discontinuity.) Almost all concepts in PT are bivalent in the sense that a concept, C, is either true or false, with no partiality of truth allowed. For example, events A and B are either independent or not independent. A process, P, is either stationary or nonstationary, and so on. An example of brittleness is: If all A's are B's and all B's are C's, then all A's are C's; but if almost all A's are B's and almost all B's are C's, then all that can be said is that the proportion of A's in C's is between 0 and 1. In particular, brittleness of independence creates

---

* Professor in the Graduate School and Director, Berkeley Initiative in Soft Computing (BISC), Computer Science Division and the Electronics Research Laboratory, Department of EECS, University of California, Berkeley, CA 94720-1776; Telephone: 510-642-4959; Fax: 510-642-1712; E-mail: zadeh@cs.berkeley.edu. Research supported in part by ONR Contract N00014-00-1-0621, ONR Contract N00014-99-C-0298, NASA Contract NCC2-1006, NASA Grant NAC2-117, ONR Grant N00014-96-1-0556, ONR Grant FDN0014991035, ARO Grant DAAH 04-961-0341 and the BISC Program of UC Berkeley

serious problems in construction of Bayesian nets since only an epsilon separates independence from dependence.

2. The dilemma of "it is possible but not probable." A simple version of this dilemma is the following. Assume that A is a proper subset of B and that the Lebesgue measure of A is arbitrarily close to the Lebesgue measure of B. Now, what can be said about the probability measure, P(A), given the probability measure P(B)? The only assertion that can be made is that P(A) lies between 0 and P(B). The uniformativeness of this assessment of P(A) leads to counterintuitive conclusions. For example, suppose that with probability 0.99 Robert returns from work within one minute of 6pm. What is the probability that he is home at 6pm? Using PT, with no additional information or the use of the maximum entropy principle, the answer is: between 0 and 1. This simple example is an instance of a basic problem of what to do when we know what is possible but cannot assess the associated probabilities or probability distributions. A case in point relates to assessment of the probability of a worst case scenario.
3. Incapability to operate on perception-based information. This is the most serious limitation of PT. It is rooted in the fact that much of human knowledge is perception-based, e.g., "Most Swedes are tall;" "It is foggy in San Francisco during the summer," "It is unlikely to be warm tomorrow." and "Usually Robert returns from work at about 6pm."

Perceptions are intrinsically imprecise, reflecting the bounded ability of sensory organs and, ultimately, the brain, to resolve detail and store information. More concretely, perceptions are f-granular in the sense that (a) the boundaries of perceived classes are unsharp; and (b) the values of perceived attributes are granulated, with a granule being a clump of values drawn together by indistinguishability, similarity, proximity or functionality.

F-granularity of perceptions puts them well beyond the reach of conventional meaning-representation methods based on bivalent logic. As a consequence, PT, by itself, lacks the capability to operate on perception-based information: As an illustration, PT cannot provide an answer to the query: What is the probability that Robert is home at about t pm, given the perception-based information that (a) usually Robert leaves his office at about 5:30 pm; and (b) usually it takes about 30 minutes to reach home.

What can be done to endow probability theory with the capability to operate on perception-based information? In a recent paper entitled "Toward a Perception-Based Theory of Probabilistic Reasoning with Imprecise Probabilities," (Journal of Statistical Planning and Inference), I outlined an approach to a restructuring of probability theory which adds this capability to PT. Briefly, the proposed restructuring involves three stages of generalization: (a) f-generalization; (b) f.g-generalization; and (c) nl-generalization. More specifically:

(a) F-generalization involves a progression from crisp sets to fuzzy sets in PT, leading to a generalization of PT which is denoted as PT^+. In PT^+, probabilities, functions, relations, measures and everything else are allowed to have fuzzy denotations, that is, be a matter of degree. In particular, probabilities described as low, high, not very high, etc. are interpreted as labels of fuzzy subsets of the unit interval or, equivalently, as possibility distributions of their numerical values.

(b) F.g-generalization involves fuzzy granulation of variables, functions, relations, etc., leading to a generalization of PT which is denoted as PT^{++}. By fuzzy granulation of a variable, X is meant a partition of the range of X into fuzzy granules, with a granule being a clump of values of X which are drawn together by indistinguishability, similarity, proximity or functionality. Membership functions of such granules are usually assumed to be triangular or trapezoidal. Basically, granularity reflects the bounded ability of the human mind to resolve detail and store information.

(c) Nl-generalization involves an addition to PT^{++} of a capability to operate on propositions expressed in a natural language, with the understanding that such propositions serve as descriptors of perceptions. Nl-generalization of PT leads to perception-based probability theory denoted as PTp. By construction, PTp is needed to answer the query in the Robert example: What is the probability that Robert is home at about t pm?

The principal thrust of what was said above may be viewed as a call for a recognition that standard probability theory has serious limitations—limitations which are rooted in its bivalent base and the fundamental conflict between bivalence and reality. What is needed to circumvent these limitations is a restructuring of the foundations of probability theory—a restructuring aimed principally at endowing probability theory with the capability to operate on perception-based information.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Polysynthetic Class Theory for the 21st Century

Edgar G. K. Lopez-Escobar

By a *Polysynthetic Class Theory* we understand a set theory in which (1) there are *sets* and *classes*, (2) the well formed expressions can be interpreted *both* as *propositions* and *classes* and (3) there is a well formed expression, $\mathcal{V}(x)$, with x as its only free variable such that:

$$\bigwedge_{x}(\mathcal{V}(x) \leftrightarrow x).$$

A. P. Morse first developed the well known impredicative set theory of the Appendix to Kelley's *General Topology* in a polysynthetic format; the polysynthetic version was finally published in the monograph *A Theory of Sets*, 1965.

Although polysynthethism violates the familiar syntax rules for post Hilbert formal languages of mathematical logic, in which the well formed expressions either correspond to classes or (in the exclusive interpretation) to propositions, it was used by the founders of the discipline, G. Boole, Jevons, E. Schröder, etc.

In this talk we extend the ideas of Boole,. . . , Morse in a formalism suitable for this century.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Development of a Fuzzy Genetic System for Data Classification

Nelson Ebecken

Abstract

This work discusses the implementation of a fuzzy genetic approach to perform data classification. A surface geochemical application is focused: how to predict the boundariesof a hydrocarbon reservoir. In other words, given the ethane concentration in the soil and certain environment data, a fuzzy classifier defines whether a specific sample site is within the reservoir boundaries. The method aims to obtain small fuzzy classifiers by means of optimization of fuzzy rules using a genetic algorithm. The main objectives are: how a fuzzy rule set is generated from the database and how its best subset can be found by the genetic algorithm. The classifiers are evaluated in terms of accuracy, cardinality and number of employed features. The method was able to produce small fuzzy classifiers and very good performance.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

Why the Problem P = NP is so Diffucult

Newton C.A. da Costa

In this paper we discuss several issues underlying the problem P = NP and its difficulties, and showing some advances concerning the theme, including some results obtained by us.

Advances in Logic, Artificial Intelligence and Robotics
J.M. Abe and J.I. da Silva Filho (Eds.)
IOS Press, 2002

The Importance of Evaluating Learning Algorithms during the Data Mining Process

Maria Carolina Monard e Gustavo E.A.P.A. Batista

Instituto de Ciências Matemáticas e de Computação -ICMC
USP-São Carlos
{mcmonard,gbatista}@icmc.sc.usp.br

We are overwhelmed with data. The amount of data in the world, in our lives, seems to go on and on increasing. More powerful computers, inexpensive storage systems, and the World Wide Web directly contribute to make enormous amounts of data available to everyone. We would all testify to the growing gap between the generation of data and our understanding of it. As the volume of data increases, inexorably, the proportion of it that people *understand* decreases alarmingly. Lying hidden in all this data is information, potentially useful information, which is rarely made explicit or taken advantage of.

Data mining is about solving problems by analyzing data already present in databases. Data mining is defined as the process of discovering patterns in data. The process must be automatic or (more usually) semi-automatic. The patterns discovered must be meaningful in that they lead to some advantage. The idea is that patters in data can be sought automatically, identified, validated, and used for prediction.

We do need experiments in data mining research because the methods we employ and the data we want to analyze are too complex for a complete formal treatment. In other words, for a given data analysis problem we do not have formal instruments to decide which of the methods is the optimal one. Of course, there is a vast literature in statistics, computational learning theory and the like that does help us in such decisions. But the last word in the decision is always spoken by an empirical check, an experiment, as in any other science that needs evaluation of its theories.

Evaluation is one of the key to making real progress in data mining. There are many methods to induce knowledge from data. However, to determine which method to use on a particular problem we need systematic ways to evaluate how different methods work and to compare one with another. Evaluation is not as simple as it might appear at first sight.

An important component of many data mining projects is finding machine learning algorithms that induce good classifiers for the specific knowledge domain under current investigation. This process requires careful thought about experimental design. Otherwise, comparative studies of classification algorithms can easily result in statistically invalid conclusions.

Our talk will concentrate in this latter topic. Several case studies will be treated in order to illustrate the importance of evaluating learning algorithms during the data mining process.

Author Index

Abe, J.M. 96
Akama, S. 96
Aoki, A.R. 38
Ávila, B.C. 144
Bassanezi, R.C. 88,208
Batista, G.E.A.P.A. 173,275
Cafezeiro, I. 216
da Costa, N.C.A. 274
da Motta Jafelice, R.S. 208
de Barros, L.C. 88,202,208
de Oliveira, R.Z.G. 88
do Amaral, F.N. 194
dos Santos, E.L. 144
Droste, M. 152,268
Ebecken, N. 273
Endler, M. 194
Esmin, A.A.A. 38
Gomide, F. 55,208
Haeusler, E.H. 194,216,224
Hasegawa, F.M. 144
Jerônimo, R.A. 250
Kaestner, C.A.A. 144
Kudo, M. 104,112,120
Kuske, D. 152
Lambert-Torres, G. 1,38,267
Lopez-Escobar, E.G.K. 272
Maldonado, E.P. 250
Manzolli, J. 46
Martins, H.G. 1
Massad, E. 68,75,258
Mendoza, N.F. 242
Menezes, R.X. 258
Miyagi, K. 9
Monard, M.C. 173,275
Mori, Y. 104,112
Moroni, A. 46
Murai, T. 126
Nakamatsu, K. 17,29
Nakao, Z. 9
Nakata, M. 126
Ortega, N.R.S. 202,258
Pereira Jr., A. 82
Pereira, J.C.R. 202
Ramos Corchado, F.F. 242
Rocha, A.F. 62,68,75
Rocha, F.T. 62
Sacchetta, T.E.P. 258
Sano, Y. 9
Sato, Y. 126
Silva, G.M.H. 194,216,224
Simão, J.M. 234
Souza, I.A. 250
Stadzisz, P.C. 234
Sueyoshi, T. 9
Suzuki, A. 17,29
Tanaka, S. 181
Tenmoto, H. 104
Thiele, H. 134
Tonelli, P.A. 202
Veloso, P.A.S. 224
Von Zuben, F.J. 46
Zadeh, L.A. 269